

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ІНТЕРНЕТ ТЕХНОЛОГІЇ

МЕТОДИЧНІ ВКАЗІВКИ
ДО ЛАБОРАТОРНИХ РОБІТ

Київ 2014

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ІНТЕРНЕТ ТЕХНОЛОГІЇ
МЕТОДИЧНІ ВКАЗІВКИ
ДО ЛАБОРАТОРНИХ РОБІТ

для студентів напрямку підготовки 6.050802
«Електронні пристрої та системи»

Київ 2014

Інтернет технології. Методичні вказівки до лабораторних робіт для студентів напрямку підготовки 6.050802 «Електронні пристрої та системи». - К.: НТУУ «КПІ», 2014. – 90 с.

Навчальне видання

Інтернет технології. Методичні вказівки до лабораторних робіт для студентів напрямку підготовки 6.050802 «Електронні пристрої та системи»

Укладачі: Кисельова Анна Геннадіївна, ст.викл, к.т.н.

Вербицький Євген Володимирович, ст.викл, к.т.н.,

Відповідальний редактор: Жуйков Валерій Якович, проф., д.т.н.

Рецензент: Чкалов Олексій Валерійович, доц., к.т.н.

Київ - 2014

ЗМІСТ

Вступ.....	4
Лабораторна робота № 1. Основи проектування веб-додатків на базі HTML-сторінки	6
Лабораторна робота № 2. Застосування основних тегів. створення посилань, робота з зображеннями.....	16
Лабораторна робота № 3. Створення таблиць і Основи CSS.....	24
Лабораторна робота № 4. CSS-властивості: розміри, кольори, шрифти, текст.....	38
Лабораторна робота № 5. Теги DIV і SPAN, псевдокласи.....	61
Лабораторна № 6. Основи верстки. Таблична верстка.....	79
Лабораторна № 7. Основи верстки. Блокова верстка	87
Література.....	90

Вступ

У комп'ютерному практикумі розглянуто основні технологічні проблеми розробки великих веб-проектів і методи вирішення цих проблем. Розглядаються технології проектування складного програмного веб-забезпечення з використання сучасних підходів.

Виконання лабораторних робіт по курсу «Інтернет-технології» надає студентам базових ґрунтовних знань щодо сучасних інтернет-технологій, та надання студентам навичок та вмінь для створення власних веб-проектів. Студенти отримають досвід практичного застосування сучасних технологій програмування веб-проектів.

Лабораторна робота № 1.

Основи проектування веб-додатків на базі HTML-сторінки

Мета роботи:

Вивчення структури HTML - сторінки і основ проектування для World Wide Web.

Короткі теоретичні відомості

Основи HTML

Мова HTML (від англ. Hyper Text Markup Language - «мова розмітки гіпертексту») служить для створення веб-сторінок. Більшість сайтів створені саме за допомогою HTML.

Синтаксис HTML

HTML-документи являють собою файли з текстом і додатковими інструкціями мови HTML, званими тегами. Теги дозволяють задавати форматування тексту, а також розміщувати в документі мультимедійні файли (зображення, звук, Flash-анімацію), гіпертекстові посилання на інші документи, табличні дані, форми введення даних. HTML-документи мають розширення імені файлу htm або html.

Редагування HTML коду виробляють в текстовому редакторі (наприклад, в звичайному блокноті), а перегляд - в браузері.

Структура тега:

<ім'я тега атрибут1 атрибут2 = "значення2" ...>

Тег складається з імені тега, за яким може слідувати список атрибутів, які розміщені між відкриває і закриває кутовими дужками (<i>). Атрибути дозволяють управляти поведінкою тега. Вони можуть мати конкретні значення, що задаються після знаку рівності. Значення атрибутів полягають в одиночні або подвійні лапки ("). Атрибути відокремлюються один від одного пробілом, порядок проходження атрибутів значення не має. Імена тегів і атрибутів нечутливі до регістру.

Приклад:

Тег FONT призначений для управління зовнішнім виглядом тексту. У прикладі він задає накреслення тексту шрифтом Arial, червоним кольором. Теги підрозділяються на парні і непарні. Парні теги мають закриває тег, непарні - не мають. Закриває тег містить косу риску перед ім'ям і не має атрибутів. Між відкриває і закриває тегами поміщається текст та інші теги. Атрибути вказуються тільки в відкриваючому тегі.

Для виділення тексту жирним використовується тег . приклад:

HTML-код: текст жирний текст </ b> текст

*У браузері: **жирний текст***

Прикладом непарного тега є тег
 - переклад рядка. Звичайний переклад рядка клавішею {Enter} браузер ігнорує (як і кілька поставлених поспіль прогалів чи знаків табуляції).

неправильно:

HTML-код:

перший рядок

другий рядок

У браузері:

перша рядок другий рядок

правильно:

HTML-код:

перший рядок
 другий рядок

У браузері:

перший рядок

другий рядок

Парний тег обов'язково повинен мати закриває! Наприклад, якщо не закрити тег , весь текст на сторінці за ним стане жирним. При вкладенні тегів один в одного закривати теги потрібно починаючи з самого останнього, у зворотному порядку.

Тег <i> використовується для виділення тексту курсивом.

Неправильно: HTML-код: <i> жирний курсив </ b> </ i>

Правильно: HTML-код: <i> жирний курсив </ i> </ b>

У браузері: жирний курсив

Структура документа HTML

заголовок документа

<html>

<head>

...

</ head>

тіло документа

<body>

...

</ body>

</ html>

HTML-документ укладений в тег <html> і складається з заголовка і тіла. Заголовок документа лежить всередині тега <head> і містить назву документа і деякі інші параметри. Тіло документа укладено в тег <body> і містить текст і теги, які повинен обробити і вивести браузер. Текст з тега <title> зазвичай

відображається в заголовку вікна браузера, а також в результатах пошуку пошукових систем.

Приклад: найпростіший HTML-документ

```
<html>
```

```
<head>
```

```
<title>Заголовок</title>
```

```
</head>
```

```
<body>
```

```
Мій перший <i> HTML-документ! </i><br>
```

```
(це приклад)
```

```
</body>
```

```
</html>
```

Подання кольору в HTML

Колір в HTML може бути заданий ключовими словами - назвами квітів на англійською мовою:

Название в HTML	Код в RGB
aqua	#00ffff
black	#000000
blue	#0000ff
fuchsia	#ff00ff
grey	#808080
green	#008000
lime	#00ff00
maroon	#800000
navy	#000080
olive	#808000
purple	#800080
red	#ff0000
silver	#c0c0c0
teal	#008080
white	#ffffff
yellow	#ffff00

Але комп'ютер може відобразити набагато більше - близько 16 мільйонів -

кольорів. Альтернативним способом завдання кольору є вказівка коду кольору в системі RGB (від англ. Red, green, blue - червоний, зелений, синій). Суть системи полягає в тому, що будь-який колір може бути представлений як змішання основних кольорів - червоного, зеленого і синього. Колір записується у вигляді 6-символьного коду.

Код являє собою шістнадцяткове число від 000000 до FFFFFFFF. Перші дві цифри відповідають червоною компоненті, наступні дві - зеленою, останні дві - синьою. Значення 00 означає повну відсутність складової, значення FF (255) - максимум складової. В якості шістнадцятирічних цифр використовуються десяткові цифри від 0 до 9 і латинські літери від A до F для позначення цифр від 10 до 15. Таким чином, виходить $256^3 \approx 16.7$ млн. Квітів - цього достатньо, щоб відтворити будь-який колір, який розрізняє людське око.

наприклад:

FF0000 - яскраво-червоний (red)

00FF00 - яскраво-зелений (green)

0000FF - яскраво-синій (blue)

000000 - чорний (black)

FFFFFF - білий (white)

Значення кольору вказується в тезі після символу решітки (#).

Наприклад для тексту:

 сірий текст </ font>

Для фону всієї сторінки в тезі body атрибут bgcolor:

<body bgcolor = "# FF0000"> жовтий фон </ body>

Значення коду складно підібрати самому, тому використовуються спеціальні інструменти.

Основні теги, робота з текстом, списки

У стандарті HTML 4.01 перераховано 91 тег. Кожен тег призначений для вирішення певної задачі: роботи з текстом, посиланнями, графікою, таблицями і т.д.

Теги структури документа

Ці теги призначені для визначення структури HTML документа і не впливають на його відображення в браузері. Тим не менш, правильно сформований документ обов'язково повинен їх утримувати.

`<html> ... </ html>` - включає в себе весь вміст веб-сторінки, в тому числі теги `<head>` і `<body>`

`<head> ... </ head>` - містить теги зі службовою інформацією про сторінку, наприклад назва в тезі `<title>`.

`<title> ... </ title>` - задає назву документа. Ця назва зазвичай відображають в заголовку вікна браузера.

`<body> ... </ body>` - зберігає вміст документа.

атрибути:

`bgcolor = "колір"` - призначає колір фону документа

`text = "колір"` - вказує колір звичайного тексту в документі

! Теги `<html>`, `<head>`, `<title>` і `<body>` задаються в документі тільки один раз!

Теги для роботи з текстом

HTML дозволяє керувати відображенням тексту на сторінці.

` ... </ b>` - виділення тексту жирним

`<i> ... </ i>` - виділення тексту курсивом

`<u> ... </ u>` - підкреслення тексту

`<sub> ... </ sub>` - формувати текст як підрядковий індекс

приклад:

HTML-код: `101 <sub> 2 </ sub> = 5`

У браузері: $101_2 = 5$

<sup> ... </ sup> - форматувати текст як надрядковий індекс

приклад:

HTML-код: 2 <sup> 8 </ sup> = 256

У браузері: $2^8 = 256$

<center> ... </ center> - вирівнювання тексту по центру

 ... </ font> - встановлює розмір, колір і гарнітуру тексту

атрибути:

color = "колір" - задає колір тексту

face = "шрифт" - визначає гарнітуру тексту; значенням атрибута може

бути список шрифтів, перерахованих через кому - в цьому випадку вибирається перший доступний шрифт

size = "1-7" - встановлює розмір шрифту (від 1 до 7)

приклад:

HTML-код:

 текст </ FONT>

У браузері: **ТЕКСТ**

<p> ... </ p> - задає початок і кінець параграфа

Атрибут:

align = "..." - визначає режим вирівнювання тексту

left - по лівому краю (за замовчуванням)

center - по центру

right - по правому краю

justify - по ширині

<hN> ... </ hN> - вкладений текст, є заголовком документа рівня N, N

приймає значення від 1 до 6. Найбільшим заголовком є <h1>, най-

меншим <h6>.

 - перенесення рядка

тег HR

<hr> - виводить горизонтальну розділову лінію

атрибути:

align = "... " - визначає режим вирівнювання лінії

left - по лівому краю

center - по центру (за замовчуванням)

right - по правому краю

noshade - використовувати суцільну лінію замість об'ємної

size = "N" - товщина лінії в пікселях

width = "N" - ширина лінії в пікселях або відсотках по відношенню до ширини екрану.

Розміри об'єктів в HTML часто вказуються в пікселях. Піксель - найменший елемент зображення на екрані (крапка). Кількість пікселів на екрані по горизонталі і вертикалі називають дозволом (наприклад, 1024 по горизонталі на 768 по вертикалі).

Робота зі списками

У HTML є можливість створювати нумеровані і маркіровані списки.

 ... </ ol> - створює нумерований список елементів

атрибути:

start = "N" - почати нумерацію з числа N

type = "... " - визначає формат нумерації

1 - арабські цифри (за замовчуванням)

A - великі літери (A, B, C)

a - малі літери (a, b, c)

I - прописні римські цифри (I, II, III)

i - рядкові римські цифри (i, ii, iii)

 ... </ ul> - створює маркований список елементів

Атрибут:

type = "..." - визначає формат маркера

disk - диск (за замовчуванням)

circle - окружність

square - квадрат

 ... </ li> - задає елемент списку у нумерованому або маркірованому списку

атрибути:

type = "..." - формат номера або маркера (див. опис і)

value = "N" - задає номер елементу списку

приклад:

HTML-код:

 арабські цифри (за замовчуванням) </ li>

<li type = "A"> прописні букви </ li>

<li type = "a"> малі літери </ li>

<li type = "I"> прописні римські цифри </ li>

<li type = "i"> рядкові римські цифри </ li>

</ ol>

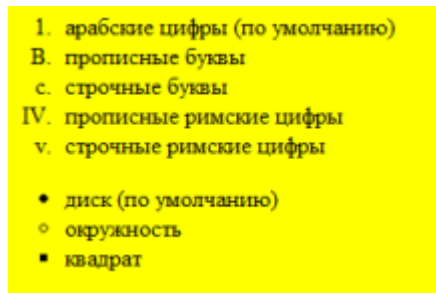
 диск (за замовчуванням) </ li>

<li type = "circle"> окружність </ li>

<li type = "square"> квадрат </ li>

</ ul>

У браузері:



Хід роботи

1) Створіть HTML-сторінку з наступним змістом: Мій перший сайт! (це приклад) Прізвище Ім'я По батькові

Назва сторінки - «Моя перша сторінка». Прізвище, ім'я та по батькові повинні виводитися різними кольорами. Назва файлу - index.html. Задайте сторінці кольоровий фон: підберіть такий колір, щоб він не утруднював читання тексту.

2) Створіть HTML-сторінку з фразою: «Кожен Мисливець Бажає Знати Де Сидить Фазан ». Кожне слово має бути відповідного кольору: червоному, оранжевому, жовтому, зеленому, блакитному, синьому, фіолетовому. дайте сторінці заголовок «Веселка».

3) Створіть сторінку «Мое хобі». Сторінка повинна містити заголовок «Мое хобі», вирівняний по центру, короткий опис вашого хобі і нуровані список ваших інтересів (спорт, науки, ігри тощо). Назва файлу - hobby.html.

4) Змініть тип нумерації на нумерацію літерами і римськими цифрами.

5) Змініть тип списку на маркований, використовуйте різні типи маркування.

6) Створіть текстовий напис великого розміру. Застосуйте до неї по черзі шрифти Wingdings, Wingdings 2, Wingdings 3 і Webdings.

Контрольні питання

1. Визначення та характеристика понять Internet, WWW, HTML

2. Базова інфраструктура Інтернет і основні протоколи.

3. Браузери: еволюція і основні сучасні сімейства.

4. Пошук інформації і його документування в контексті розробки тематичного сайту.

5. Основні пошукові системи, орієнтовані на різні мовні простору .:
6. Основні характеристики відкритого і прихованого інформаційного веб-простору
7. Модель веб-простору Брєдера (Bow Tie) та її властивості.
8. Передумови появи і еволюція гіпертексту.
9. Основні етапи у розвитку HTML.

Лабораторна робота 2.

Застосування основних тегів. створення посилань, робота з зображеннями

Мета роботи

Вивчення основних тегів, що використовуються при проектуванні HTML-сторінок.

Короткі теоретичні відомості створення посилань

Для створення посилань використовується тег `<a> ... `.

Обов'язковий атрибут `href` вказує абсолютний або відносний перенос, на який веде посилання. Посилання може вказувати на HTML-документ, зображення, файл для збереження на диск та ін. Текст посилання записується між відкриває і закриває тегом.

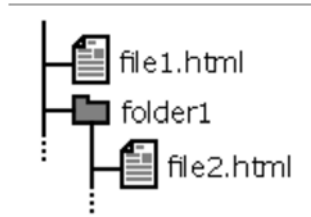
Абсолютний адреса містить в собі ім'я хоста і повний шлях до ресурсу, наприклад: `http://www.example.com/docs/about.html`. За допомогою абсолютного адреси можна посилатися на будь-який відкритий ресурс в Інтернеті. якщо потрібно поставити посилання на головну сторінку сайту, вказують його адресу слеш.

Приклад для абсолютної адреси:

HTML-код: ` Яндекс `

У браузері: Яндекс

У браузері посилання зазвичай представляється як підкреслений текст. При кліке за посиланням браузер завантажує сторінку, зазначену в атрибуті href. Також для документів, розташованих на тому ж сайті, можна використовувати відносний адресу.



Малюнок 2.1. Приклад файлової структури

Наприклад, щоб поставити посилання з файлу file1.html на файл file2.html (див. рис. 3.1), необхідний наступний HTML-код:

```
<A Href="folder1/file2.html"> файл file2.html </A>
```

А щоб посилання у файлі file2.html вказувала на file1.html:

```
<A Href="../file1.html"> файл file1.html </A>
```

Дві точки (..) означають перехід до батьківського каталогу.

Зауваження: для файлів в межах одного сайту рекомендується використовувати тільки відносні шляхи. У цьому випадку посилання збережуть работоспособність при зміні адреси сайту, перенесення в іншу папку і т.п.

Для відкриття посилання в новому вікні використовується атрибут target зі значенням _blank.

Приклад: ` Яндекс `

Колір посилань у документі можна вказати атрибутами тега <a>:

alink = "колір" - встановлює колір активних посилань

link = "колір" - задає колір невідвіданих посилань

vlink = "колір" - визначає колір відвіданих посилань зображення

Вставка зображень на сторінці

Здійснюється непарним тегом . Обов'язковий атрибут src вказує абсолютний або відносний URL зображення. Стандартними форматами зображень є GIF, PNG і JPEG.

GIF - формат, який реалізує стиснення без втрати якості з обмеженою кольоровістю (від 2 до 256 кольорів) і підтримкою анімації - використовується для зберігання графіки, коли досить 256 (і менше) кольорів. Зазвичай це невеликі зображення. Також GIF підтримує прозорість. JPEG реалізує стиснення зображень з втратами якості, при цьому обмеження на колір відсутні (підтримується 16 мільйонів кольорів). Розмір JPEG-файла залежить від параметра «якість», який вказується при його збереженні: від 0 до 100. Чим вище якість, тим більше розмір файлу. оптимальна ступінь якості залежить від зображення, в більшості випадків вона дорівнює 70-80. Не варто виставляти цей параметр менше 50 - на зображенні з'являться помітні дефекти або більше 95 - розмір файлу сильно зросте без видимого поліпшення якості.

Формат PNG існує у двох варіантах: PNG-8 і PNG-24. PNG-8, як і GIF, підтримує 256 кольорів, забезпечує в порівнянні з ним краще стиск, але не підтримує анімацію. Формат PNG-24, як і JPEG, що не має обмежень на кількість кольорів, але програє йому в розмірі файлу.

Здійснює стиснення зображень без втрати якості, тому його варто застосовувати для зображень, що містять дрібні деталі.

Нижче наведена таблиця оптимального вибору формату файлу в відносині якість / розмір файлу з урахуванням особливостей зображення.

Таблиця 1. Вибір формату графічного файлу

особливості зображення	Формат
анімоване зображення	GIF
маленьке зображення з невеликою кількістю кольорів	GIF чи PNG-8
зображення з напівпрозорістю	PNG
зображення з великою кількістю квітів, наприклад фотографія	JPEG
зображення з великою кількістю квітів	PNG-24

з дрібними деталями, наприклад скріншот (знімок екрану)	
---	--

Уникайте використання інших форматів зображень (наприклад, BMP або TIFF), т.к. вони можуть не підтримуватися окремими типами браузерів.

Інші атрибути:

`align = "..."` - визначає режим вирівнювання зображення щодо тексту в рядку:

`top` - по верхньому краю

`middle` - по центру рядка

`bottom` - по нижньому краю (за замовчуванням)

`left` - по лівому краю вікна

`right` - по правому краю вікна

`alt = "..."` - визначає текст, що описує зображення для браузерів без підтримки графіки (або з відключеною графікою), пошукових машин і т.п.

`border = "N"` - встановлює товщину рамки навколо зображень, рівної N пікселів, 0 - для відключення рамки

`height = "N"` - висота зображення в пікселях або відсотках

`width = "N"` - ширина зображення в пікселях або відсотках

Браузер визначає розмір зображення автоматично. Для прискорення завантаження рекомендується зазначати розмір зображення атрибутами `height` і `width`, щоб браузер не обчислював цей розмір автоматично після завантаження зображення. Також цими атрибутами можна розтягнути / стиснути зображення по горизонталі / вертикалі, але таке масштабування призведе до втрати якості. Зображення може бути зроблено посиланням, шляхом приміщення всередину тега `<a>`. У цьому випадку навколо зображення автоматично з'являється рамка. Товщина рамки задається атрибутом `border`. Зазвичай рамку прибирають, вказуючи

`border = "0"` в тезі ``.

приклади:

1. HTML-сторінка знаходиться в папці site, а зображення picture.jpg знаходиться в папці site / images /.

```
<img src = "images / picture.jpg" alt = "фотографія">
```

2. Зображення знаходиться на іншому сайті в Інтернет

```
<img src = "http://example.com/pics/tree.gif" alt = "дерево">
```

Фонове зображення сторінки

Можна задавати адресу фонового зображення для сторінки в атрибуті background тега <body>. Фонове зображення відображається в натуральну величину. Якщо розмір зображення менше розміру вікна браузера, то малюнок повторюється по горизонталі вправо і по вертикалі вниз. наприклад, задамо фоновим зображенням сторінки рисунок bg1.jpg.

Тест!

Рис. 2.2. Файл bg1.jpg

```
<html>
<head>
<title>Тест фона</title>
</head>
<body background="bg1.jpg">
</body>
</html>
```

В браузері:

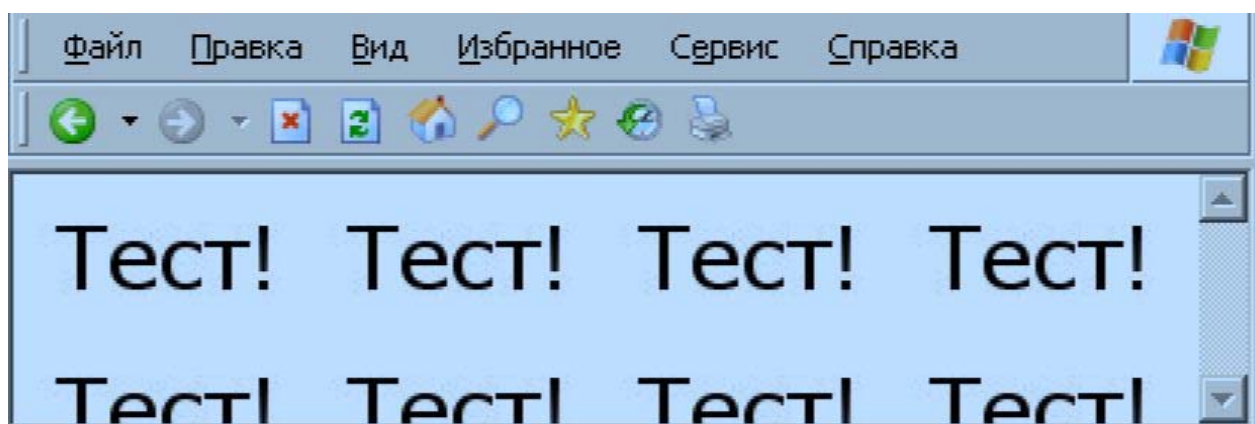


Рисунок 2.3. Розмноження фонового рисунку в браузері

Щоб не повторювався фон, необхідно вибрати картинку свідомо більшу, ніж розмір сторінки по ширині і висоті.

Можна використовувати повторюваний фон:



Рисунок 2.4. «Шахи» фон

```
<html>
<head>
<title>Тест фона №2</title>
</head>
<body background="checkmate.gif">
Сторінка з фоном «шахи».
</body>
</html>
```

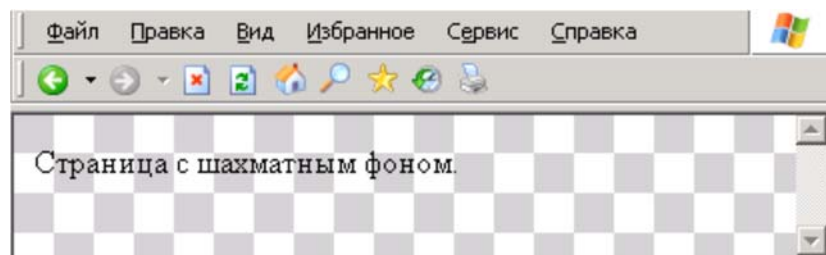


Рисунок 2.5. Використовування фона «шахи»

Якщо взяти картинку шириною 1 піксель і висотою, наприклад, 2000 пікселів на екрані вона буде розмножуватися тільки по горизонталі.

```
<html>
<head>
<title>Тест фона</title>
```

```

</head>
<body background="1px.gif">
<h1><font color="white" face="Arial">Фон</font></h1>
Використовуємо картинку з градієнтом!
</body>
</html>

```

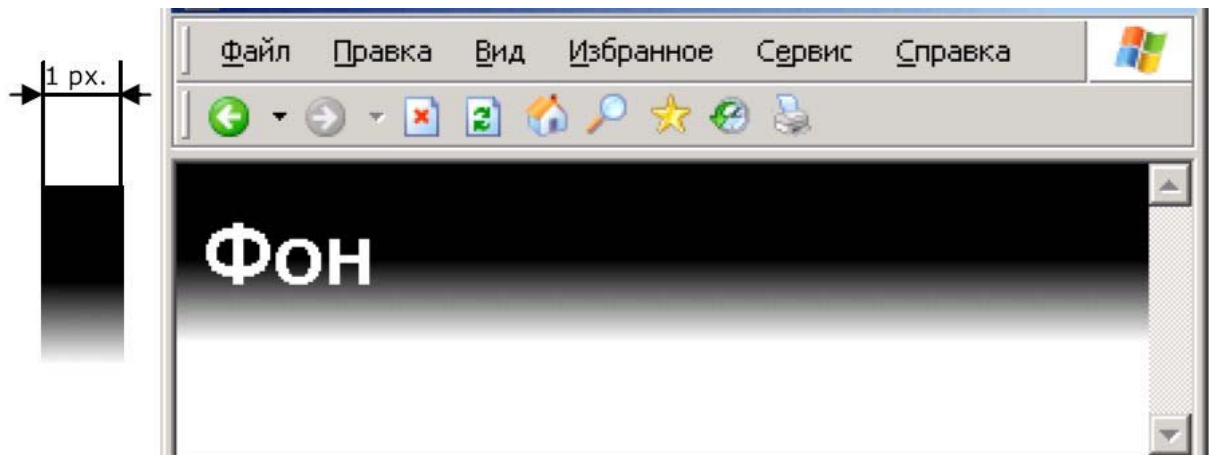


Рисунок 2.6. використання зображення, яке повторюється по горизонталі 1 px.

При використанні зображень як фону важливо забезпечити читабельність тексту на сторінці. Тому не варто застосовувати фотографії в якості фонових картинок.

Хід роботи

- 1) Модифікуйте файл index.html: додайте посилання на сторінку «Моє хоббі » і посилання на сайт відділу (повинна відкриватися в новому вікні). на сторінці «Моє хобі» додайте гіперпосилання, яке вказує на сторінку index.html.
- 2) Створіть сторінку links.html. Розмістіть на ній посилання на ваші любимі сайти. Посилання повинні бути розташовані в нумерованому або маркованому списку і відкриватися в новому вікні.
- 3) Додайте на першу сторінку (index.html) свою фотографію замість рядків «Мій перший сайт, це приклад».
- 4) За допомогою атрибутів width і height зменшіть і збільште розмір зображення в 2 рази. Зверніть увагу на втрату якості зображення при збільшенні.

- 5) Зробіть зображення на першій сторінці гіперпосиланням: при натисканні на фотографію повинен відкриватися повнорозмірний варіант в новому вікні.
- 6) Додайте графічний фон на сторінки сайту.
- 7) Додайте на сторінку інформер (невелика картинка, що показує погоду, курс валют і т.п. актуальну інформацію). URL інформера можна знайти в пошуковій системі або на спеціалізованому сайті. наприклад: <http://www.informer.ru/>, <http://gismeteo.ru/>

Контрольні питання

- 1) Модель і базова структура HTML-документів.
- 2) Основні вимоги до заголовної частини HTML.
- 3) Основні теги HTML для роботи з текстом.
- 4) Основні теги HTML для роботи із зображеннями.
- 5) Основні теги HTML для роботи з посиланнями.
- 6) Основні теги HTML для роботи зі списками.
- 7) Основні теги HTML для роботи з таблицями.

Лабораторна робота 3.

Створення таблиць і Основи CSS

Мета роботи

Вивчення властивостей елементів і особливостей їх застосування при проектуванні HTML - сторінок.

Короткі теоретичні відомості

Таблиця в HTML - це сукупність даних, розташованих і пов'язаних між собою за допомогою комірок, що розміщуються в рядках і колонках. Таблиця заповнюється даними построчно. Для вставки таблиць визначено 3 основних тега. Вміст комірок поміщається в теги `<td> ... </td>`, які, в свою чергу, поміщаються в теги рядків `<tr> ... </tr>`, а вони вже - в тег `<table> ... </table>`.

приклад:

```
<table>  
  
<tr> <td>1</td> <td>2</td> <td>3</td> </tr>  
  
<tr> <td>4</td> <td>5</td> <td>6</td> </tr>  
  
</table>
```

В браузері:

1	2	3
4	5	6

Кількість тегів `<tr> ... </tr>` визначає кількість рядків. У кожному тезі рядки повинно бути одне і те ж число тегів `<td> ... </td>`, рівне числу стовпців, інакше таблиця відобразиться неправильно.

Можна створювати вкладені таблиці: вкладати таблицю в комірку іншої таблиці.

`<table> ... </table>` - визначає початок і кінець коду таблиці, містить в собі теги рядків і осередків.

атрибути:

`align = "..."` - визначає режим вирівнювання таблиці щодо тексту в рядку

`left` - по лівому краю

`right` - по правому краю

`background = "URL"` - задає фоновий малюнок в таблиці

`bgcolor = "колір"` - колір фону таблиці

`border = "N"` - встановлює товщину меж таблиці, рівну N пікселів (0 для відключення)

`bordercolor = "колір"` - колір рамки

приклад:

`cellpadding="0"`

1	2
3	4

`cellpadding="15"`

1	2
3	4

`cellspacing = "N"` - розмір вільного простору між комірками

приклад:

`cellspacing="0"`

`cellspacing="15"`

1	2
3	4

1	2
3	4

`width = "N"` - ширина таблиці в пікселях або відсотках від ширини вікна

`<tr> ... </tr>` - визначає рядок елементів таблиці

атрибути:

`align = "..."` - визначає режим вирівнювання вмісту осередків рядка

`left` - по лівому краю

`center` - по центру

`right` - по правому краю

`justify` - по ширині

`background = "URL"` - URL зображення, яке заповнить фон осередків рядка

`bgsolor = "колір"` - колір фону комірок рядка

`valign = "..."` - визначає режим вирівнювання вмісту осередків рядка по вертикале

`top` - по верхньому краю

`middle` - по середині (за замовчуванням)

`bottom` - по нижньому краю

`<td> ... </td>` - визначає осередок даних таблиці

атрибути:

`align = "..."` - визначає режим вирівнювання вмісту комірки

`left` - по лівому краю

`center` - по центру

`right` - по правому краю

background = "URL" - URL зображення, яке заповнить фон осередку

bgcolor = "колір" - колір фону комірки

valign = "..." - визначає режим вирівнювання вмісту комірки по вертикалі

top - по верхньому краю

middle - по середині (за замовчуванням)

bottom - по нижньому краю

height = "N" - висота комірки в пікселях

width = "N" - ширина комірки в пікселях або відсотках від ширини таблиці

об'єднання осередків

colspan = "N" - розтягує осередок на N стовпців вліво

приклад:

HTML-код:

```
<TABLE cellpadding="15" border="1">
```

```
<TR><TD colspan="2">1</TD></TR>
```

```
<TR><TD>2</TD><TD>3</TD></TR>
```

```
</TABLE>
```

В браузері:

1	
2	3

rowspan="N" – растягивает ячейку на N строк вниз

Пример:

HTML-код:

```
<TABLE cellpadding="15" border="1">
```

```
<TR><TD rowspan="2">1</TD><TD>2</TD></TR>
```

```
<TR><TD>3</TD></TR>
```

```
</TABLE>
```

В браузері:

1	2
	3

Ширина таблиці

Якщо ширина таблиці спочатку не задана, то вона обчислюється виходячи з вмісту осередків.

```
<table border = "1">
```

```
<tr> <td> Ширина таблиці не задана! </ td> </ tr>
```

```
</ table>
```

Ширина таблицы не задана!

Максимальна ширина таблиці в такому випадку дорівнює ширині вікна. Якщо ж ширина задана атрибутом `width`, то браузер розставляє переноси слів у тексті осередків таким чином, щоб дотримати заданий розмір.

```
<table width = "100" border = "1">
```

```
<tr> <td> Якщо задати атрибут width, текст починає переноситися за словами </  
td> </ tr>
```

```
</ table>
```

Если задать
атрибут
width, текст
начинает пе-
реноситься
по словам

Основи CSS

CSS (Cascading Style Sheets - каскадні таблиці стилів, вимовляється «сієсс») - технологія управління зовнішнім виглядом елементів (тегів) вебсторінки. CSS надає набагато більше можливостей по оформленню сторінки, ніж HTML. Наприклад, за допомогою стилів CSS можна прибрати у посилань підкреслення, зробити у таблиці пунктирні кордону або навіть поміняти курсор «миші». Зараз CSS використовується практично на всіх сайтах Всесвітньої павутини.

Синтаксис CSS

Розглянемо синтаксис CSS. У стилях задається набір правил відображення в парах «властивість - значення», і те, до яких елементів їх застосовувати (селектор):

```
селектор
{
    свойство1: значення1;
    властивість2: значення2;
    свойство3: значеніє3 значеніє4;
}
```

Правила записуються всередині фігурних дужок і відокремлюються один від одного крапкою з комою. Між властивостями і їх значеннями ставиться двокрапка.

CSS, як і HTML, ігнорує пробіли. Можна додавати коментарі, за-
ключая їх між / * і * /.

селектори

Селектор визначає, до яких елементів (тегами) сторінки будуть застосовуватися правила, задані парами «властивість - значення».

В якості селектора можна використовувати:

1) Назва тега - тоді стиль застосовується до всіх таких тегів.

приклад:

```
A {font-size: 12pt; text-decoration: none}
```

```
TABLE {border: black solid 1px}
```

Перший рядок цього CSS-коду задає всіх посиланнях 12-й розмір шрифту і прибирає підкреслення. На другій сходинці вказується, що у всіх таблиць кордон буде чорного кольору, суцільний (solid) і шириною 1 піксель.

2) Кілька тегів через кому - тоді стиль застосовується для всіх перерахованих тегів.

приклад:

```
H1, H2, H3, H4, H5, H6 {color: red} / * робимо всі заголовки червоними * /
```

3) Кілька тегів через пропуск:

```
TABLE A {font-size: 120%}
```

Правило відноситься до всіх тегів А, вкладених в тег TABLE. Шрифт збільшиться на 20% від базового.

4) ID елемента. У стилях унікальний ідентифікатор вказується після знака # - правила застосовуються до тегу з атрибутом id = "ідентифікатор".

приклад:

CSS

```
#supersize {font-size: 200%}
```

HTML

` Довідник HTML і CSS `

Не можна вносити в документ кілька елементів з однаковим id!

5) Символ * - правила застосуються до всіх елементів документа.

6) Класи

класи

Часто потрібно, щоб стиль застосовувався до всіх тегам на сторінці, а тільки до деяких елементів (наприклад, не до всіх посилань на сторінці, а тільки до тих, які розташовані в меню сайту). Для цього використовуються класи:

ТЕГ.ім'я_класа {...}

Правила, зазначені після такого селектора, будуть діяти лише на теги з атрибутом `class = "ім'я_класу"`:

`<ТЕГ class = "ім'я_класу"> ... </ ТЕГ>`

Можна не вказувати ім'я тега, тоді правила будуть застосовуються до всіх тегам з відповідним значенням атрибута `class`.

Розглянемо приклад:

Для всіх тегів з атрибутом `class = "class1"` додамо підкреслення тексту і зменшимо розмір шрифту, а для тега `` приберемо підкреслення.

`.class1 {text-decoration: underline; font-size: 80%;}`

`A.class1 {text-decoration: none;}`

У HTML-кодi вкажемо для тегів ім'я класу:

`<h1 class = "class1"> Мої улюблені сайти </ h1>`

``

Яндекс `
`

``

Google `
`

` Redut.ru `

У браузері буде відображатися:

Мої улюблені сайти

Яндекс

Google

Redut.ru

Можна вказувати для одного елемента кілька класів через пробіл.

Стили CSS можуть включатися в HTML-документ 3 різними способами:

Зовнішні стилі.

Зберігаються в окремому файлі .css. Підключаються тегом `<link rel = "stylesheet" type = "text / css" href = "адрес_стіля">`

Основна перевага: один стиль може використовуватися відразу в деяких документах HTML.

У зовнішніх файлах потрібно зберігати стилі, загальні для всього сайту, вони впливають відразу на безліч тегів у безлічі документів. Це стає дуже зручним, якщо сайт містить багато сторінок. Наприклад, ми хочемо поміняти на всіх сторінках сайту колір фону і розмір шрифту. Якщо всі сторінки підключають один і той же зовнішній стиль CSS, достатньо в ньому задати новий колір фону і розмір шрифту. Інакше доведеться редагувати кожен документ окремо. Якщо на сайті кілька десятків або сотень сторінок це стає дуже трудомістким завданням.

CSS-файл може знаходитися і на іншому сайті - у цьому випадку необхідно вказати його абсолютний URL-адресу.

Реалізуємо наш попередній приклад. Створимо файл style.css:

```
.class1 {text-decoration: underline; font-size: 80%}
```

```
A.class1 {text-decoration: none;}
```

Тепер створимо саму сторінку links.html:

```
<html>
```

```
<head>
```

```
<link rel = "stylesheet" type = "text / css" href = "style.css">
```

```
</ head>
```

```
<body>
```

```
<h1 class = "class1"> Мої улюблені сайти </ h1>  
<a href="http://yandex.ru" class="class1">  
Яндекс </a> <br>  
<a href="http://google.com" class="class1">  
Google </a> <br>  
<a href="http://redut.ru" class="class1"> Redut.ru </a>  
</ body>  
</ html>
```

При відкритті цієї сторінки браузер клієнта завантажить також файл style.css і застосує правила CSS до документа.

Зверніть увагу: за допомогою CSS можна відключити у посилань підкреслення. Засобами HTML цього зробити неможливо. CSS значно розширює можливості оформлення сторінки.

Другий важливий момент: використання CSS дозволяє розділити оформлення та вміст документа. У нашому прикладі правила оформлення містяться у файлі style.css, а зміст - в links.html. Такий поділ істотно спрощує редагування сайту в подальшому. Рекомендується для оформлення використовувати тільки засоби CSS, відмовитися від використання таких тегів, як , <s>, <u>, <center>, атрибутів align, border, color, height, width і т.д.

Стилі рівня документа

Застосовуються до всього документа, записуються всередині тега <style> ... </style>, який вкладається в тег <head> ... </ head> в документі HTML.

Такий спосіб вказівки стилів використовується, коли потрібно застосувати однакові стилі відразу до безлічі HTML-елементів (тегів) в одному документі.

Додамо в наш приклад тег <style>:

```
<html>  
<head>  
<link rel = "stylesheet" type = "text / css" href = "style.css">
```



```
<style>

</ style>
</ head>

<body>
<h1 class = "class1"> Мої улюблені сайти </ h1>
<a href="http://yandex.ru" class="class1">
Яндекс </a> <br>
<a href="http://google.com" class="class1">
Google </a> <br>
<a href="http://redut.ru" class="class1"> Redut.ru </a>

</ body>
</ html>
```

внутрішні стилі

Використовуються, коли потрібно вказати стилі конкретного єдиного елемента. Внутрішній стиль записується в атрибуті style і застосовується тільки до вмісту цього тега. Внутрішній стиль має більш високий пріоритет, ніж зовнішні стилі і стиль рівня документа. Переважно не використовувати такий спосіб завдання стилю, тому він не відповідає принципу поділу змісту та оформлення.

Порядок застосування стилів

При роботі з CSS необхідно пам'ятати, що більш специфічні правила мають пріоритет над менш специфічними, наприклад: стиль, вказаний в атрибуті style, перекриває стиль, вказаний в тезі <style> або зовнішньому файлі CSS:

```
<html>

<head>

<style>
```

```
A {color: red; text-decoration: none}  
</ style>  
</ head>  
<body>  
<a href=http://intuit.ru style="color: green"> INTUIT </a>  
</ body>  
</ html>
```

У браузері посилання буде не підкреслене, зеленого кольору. селектор ID (#) має більший пріоритет, ніж селектор класу (.), а той, у свою чергу, - більший, ніж звичайний селектор тега:

```
<html>  
<head>  
<style>  
A {color: red; text-decoration: none; font-size: 120%}  
.links {color: blue; text-decoration: underline}  
#greenlink {color: green}  
</ style>  
</ head>  
<body>  
<a href = "http://htmlbook.ru" class = "links"  
id = "greenlink"> htmlbook.ru </a>  
</ body>  
</ html>
```

У браузері посилання буде зеленою і підкресленою, розмір шрифту збільшений на 20%.

Іншою важливою особливістю CSS є те, що деякі атрибути успадковуються від батьківського елемента до дочірнього. Наприклад, якщо атрибут font-size заданий для тега <body>, то він успадковується всіма елементами на сторінці. Коли властивість розміру задається у відсотках, воно буде обчислено виходячи

із значення для батьківського елемента. Дізнатися, чи є атрибут спадкоємною, можна в довіднику по атрибутам CSS (наприклад, <http://htmlbook.ru>).

Хід роботи

а) Створіть наступну таблицю:

1	2	
	3	4

2) Додайте сторінку «Мій комп'ютер» (computer.html), що містить заголовок сторінки і таблицю даних про вашому комп'ютері (апаратне забезпечення: процесор, об'єм оперативної пам'яті і т.п., операційна система) і посилання на головну сторінку.

Додайте посилання на computer.html на головній сторінці.

Параметри процесора і пам'яті можна подивитися у властивостях піктограми «Мій комп'ютер», обсяг жорстких дисків - в «Моєму комп'ютері».

приклад:

Процессор	Intel Core2 Duo 2.66 ГГц
Оперативная память	2 Гб
Жесткий диск	250 Гб
Подключение к интернет	10 Мбит/с
Операционная система	Windows XP Professional

3) Створіть таблицю основних кольорів палітри RGB та їх комбінацій. Жовтий, пурпурний і бірюзовий кольори виходять при змішуванні червоного, зеленого і синього один з одним. Фон кожного осередку повинен відповідати зазначеному кольором.

-	Красный FF0000	Зеленый 00FF00	Синий 0000FF
Красный FF0000	Красный FF0000	Желтый FFFF00	Пурпурный FF00FF
Зеленый FF0000	Желтый FFFF00	Зеленый 00FF00	Бирюзовый FF00FF
Синий 0000FF	Пурпурный FFFF00	Бирюзовый 00FF00	Синий FF00FF

- 4) Створіть зовнішній CSS файл. Підключіть його до всіх сторінок вашого сайту. Збільшіть розмір шрифту, задайте для тега BODY фон властивістю background-color і кордон товщиною 5px.
- 5) На головній сторінці змініть колір фону на відмінний від кольору на інших сторінках.
- 6) Створіть 2 різних класу стилів для посилань на внутрішні сторінки (в навігаційному меню) і зовнішніх посилань. Додайте атрибут class в теги <a> на сторінках.
- 7) Збережіть на диск копію якої-небудь сторінки з Інтернету. Відредагуйте її код: додайте кордон для всіх елементів сторінки.

Контрольні питання

- 1) Основні способи створення HTML файлів.
- 2) URL - призначення та структура.
- 3) Класифікація гіперпосилань за різними критеріями.
- 4) Статистика поширення основних мов, індекси цитування та «індекс мовної ефективності» у веб-просторі.
- 5) Методика перекладу текстів на базі використання ресурсів Інтернет і використовуваних при цьому інструменти.

Лабораторна робота 4

CSS-властивості: розміри, кольори, шрифти, текст

Мета роботи

Вивчення стилів, використовуваних при проектуванні HTML - сторінок.

Короткі теоретичні відомості

CSS-властивості: розміри, кольори, шрифти, текст

розміри

Розміри в CSS можна задавати в різних одиницях виміру:

em - поточна висота шрифту

pt - пункти (друкарський одиниця виміру шрифту)

px - піксель

% - Відсоток

Набагато рідше використовується вказівка розмірів в міліметрах (mm), сантиметрах (cm) і дюймах (in).

Одиниця виміру записується відразу за значенням без пробілу:

TABLE {font-size: 12pt}

кольори

У CSS колір задається як і в HTML - 6 шістнадцатерічними цифрами: по 2 на кожен базовий колір (червоний, зелений, синій). Також можна використовувати стандартні назви кольорів англійською.

наприклад:

A.content {color: black}

A.menu {color: # 3300AA}

Допускається скорочувати шістнадцаткове представлення до 3 цифр: запис # 3300AA можна замінити на # 30A.

Рідше використовується конструкція rgb (...), яка дозволяє задавати червону, зелену і синю компоненти в десятковому або відсотком вигляді:

A.content {color: rgb (0%, 0%, 0%)}

A.menu {color: rgb (51,0,170)}

URL

URL задаються конструкцією url (...). Наприклад, наступний CSS-код додає фонове зображення для сторінки:

BODY {background-image: url (images / bg.jpg);}

шрифти

Шрифт - набір накреслень літер і знаків. У комп'ютері шрифт є файл, в якому описано, як повинні відображатися на моніторі або принтері різні символи: літери, цифри, знаки пунктуації та ін. Часто шрифти містять тільки накреслення для латинського алфавіту і не мають, наприклад, підтримки кирилиці. Існують Unicode-шрифти, які містять символи для всіх мов. Основні формати файлів шрифтів: TTF - TrueType і його розширення OTF - OpenType.

serif - шрифти із зарубками (антіквенние), наприклад: Times New Roman, Georgia.

sans-serif - рубані шрифти (шрифти без зарубок або гротески), типовими представники - Arial, Impact, Tahoma, Verdana;

cursive - курсивні шрифти: Comic Sans MS;

fantasy - декоративні шрифти, наприклад: Curlz MT.

monospace - моноширинних шрифти, ширина кожного символу однакова.

Приклади: Courier New, Lucida Console.

Зарубками називають елементи на кінцях штрихів букв. порівняємо букву шрифту Times New Roman і букву шрифту Arial.

Типи шрифтів:



Рис. 4.1. Порівняння букви М різним шрифтом.

Пунктирними лініями обведені зарубки.

Використання шрифтів із зарубками полегшує читання тексту з паперу, при цьому такі шрифти зазвичай використовують для набору основного тексту в книгах. Для web-сайтів основний текст частіше набирають шрифтом без зарубок:

Arial, Tahoma, Trebuchet MS, Verdana.

текст

CSS дозволяє управляти властивостями шрифту і тексту. `font-family` - задає накреслення шрифту. Можна вказати кілька значень ний через кому. Браузер перевірить перший шрифт зі списку: якщо шрифт встановлений на комп'ютері користувача, то браузер застосує його, якщо не перейде до другого шрифту і т.д. Останнім у списку зазвичай вказується загальний тип шрифту `serif`, `sans-serif`, `cursive`, `fantasy` або `monospace`.

приклад:

`font-family: Georgia, 'Times New Roman', serif`

Якщо на комп'ютері користувача встановлений шрифт Georgia, то буде використовуватися він, якщо ні - то Times New Roman. Якщо ж і Times New Roman відсутня, то браузер буде використовувати шрифт із зарубками, який встановлений на комп'ютері.

Ще в CSS2 була реалізована підтримка методу `fontface` для завантаження відсутніх шрифтів з сервера, але до недавнього часу не всі браузери підтримували цю можливість. Зараз `fontface` працює в останніх версіях FireFox, Opera, Safari. Internet Explorer реалізує `fontface` з 4 версії, але підтримує тільки шрифти в форматі EOT (Embedded OpenType), які можуть бути отримані з TrueType і OpenType програмою-конвертером.

font-size - розмір шрифту. Може здаватися абсолютним значенням в пунктах (pt) або пікселях (px) або відносним - у відсотках (%) або в em.

приклад:

font-size: 12pt

або

font-size: 150%

font-style - задає накреслення тексту: normal (звичайне), italic (курсивное) або oblique (похиле). Курсивне зображення є спеціальною зміненою версією шрифту, що імітує рукописний текст з нахилом вправо. Похиле накреслення виходить із звичайного нахилом букв. Різниця видно на прикладі:

The five boxing wizards jump quickly.
The five boxing wizards jump quickly.
The five boxing wizards jump quickly.

Рисунок 4.2. Нормальний, курсивний і похиле накреслення.

Зазвичай браузер не може відобразити похиле накреслення і замінює його курсивним.

font-weight - дозволяє змінити рівень жирності тексту: normal (звичайна), bold (напівжирний). Дія аналогічно тегу .

У специфікації CSS 2.1 визначені й інші значення властивості fontweight крім normal і bold, але на даний момент браузери погано їх підтримують.

color - задає колір тексту.

Наприклад, задамо червоний колір для всіх заголовків:

H1, H2, H3, H4, H5, H6 {color: # ff0000}

або

H1, H2, H3, H4, H5, H6 {color: red}

line-height - міжрядковий інтервал (інтерліньяж), вказує відстань між рядками тексту. Може здаватися числом як множник від поточного розміру шрифту, у відсотках, а також у пунктах (pt), пікселях (px) і других одиницях виміру CSS.

приклад:

line-height: 1.5; / * Полуторний інтервал * /

У програмуванні прийнято відокремлювати цілу частину числа від дробової точкою, як в англійській мові. У російській мові використовується кома.

text-decoration - задає оформлення тексту. Варіанти: line-through (перекреслений), overline (лінія над текстом), underline (підкреслення), none (відключення ефектів).

Наприклад, відключимо підкреслення у посилань:

A {text-decoration: none}

text-align - вирівнювання тексту в блоці: left (по лівому краю), center (по центру), right (по правому краю) або justify (по ширині).

приклад:

P {text-align: justify}

text-indent - відступ першого рядка («новий рядок»). Довжина відступу може здаватися у відсотках (%) від ширини текстового блоку, пікселях (px), пунктах (pt) та ін.

приклад:

P {text-indent: 1.25cm}

Властивості font-style, font-variant, font-weight, font-size, font-family і line-height можна задати в одному правилі:

font: font-style font-weight font-size / line-height font-family

Значення font-size і font-family є обов'язковими, інші можна не вказувати, наприклад:

H1 {font: bold 14pt / 1.5 sans-serif}

CSS-властивості: поля, заповнення, межі

У CSS кожен елемент розташовується в блоці, якому можна задати значення полів (margin), заповнення (padding) і межі (border). поле є відступом елемента від сусідніх, а заповнення - порожній областю між границею і вмістом (див. рис. 4.3).

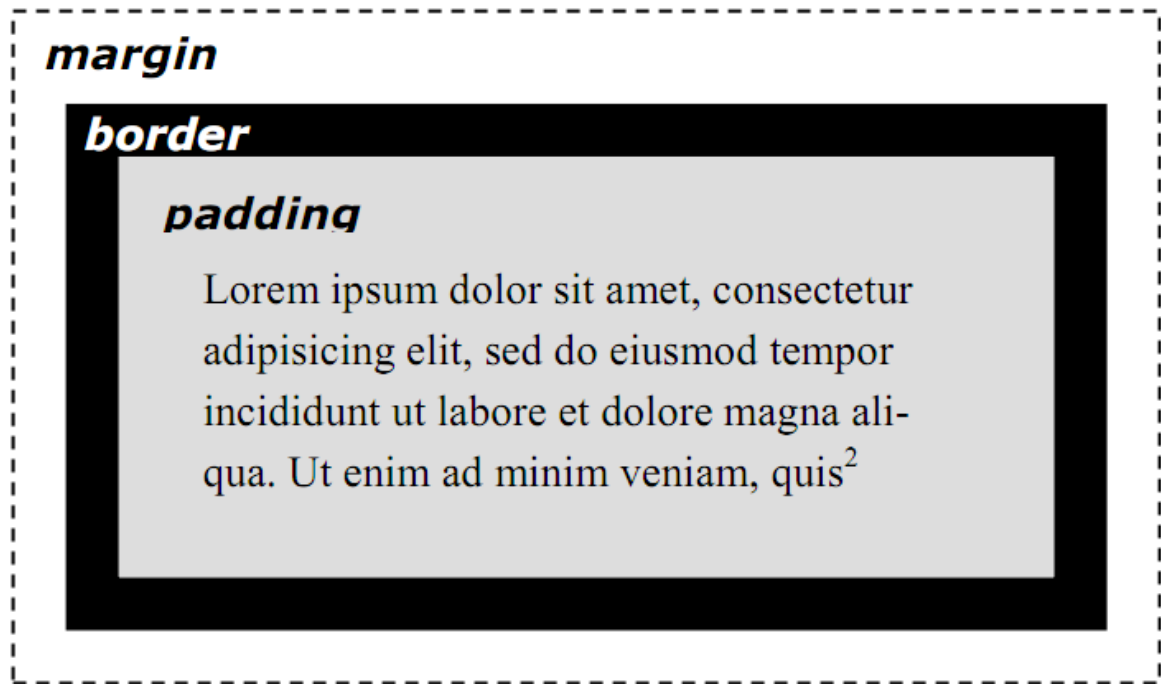


Рис 4.3 Бокс (box) елемента.

Ширина полів і заповнення задається наступними CSS властивостями: margin-top, margin-right, margin-bottom, margin-left - для верхньої, правої, нижньої, лівої сторони поля.

margin - скорочений запис. Задає значення відразу для всіх сторін.

приклад:

```
P {margin: 10px}
```

аналогічно запису

```
P {
```

```
margin-top: 10px;
```

```
margin-right: 10px;
```

```
margin-bottom: 10px;
```

```
margin-left: 10px;
```

```
}
```

Якщо для `margin` вказати два значення через пропуск, то перше з них буде задавати ширину верхнього і нижнього поля, а другий - лівого і правого. Якщо вказати три значення, то перше буде присвоюватися верхньому полю, друге - лівому і правому, а третє - нижньому. Нарешті, при вказівці чотирьох значень, вони по черзі будуть вказувати верхнє, праве, нижнє і лівє поля.

`padding-top`, `padding-right`, `padding-bottom`, `padding-left` - встановлюють ширину заповнення зверху, справа, знизу і зліва від вмісту відповідно.

`padding` - встановлює значення відразу для всіх сторін.

`Padding` може приймати не тільки одне, але і 2, 3 або 4 значення. См. примітка для `margin`.

Для `margin` і `padding` можна задавати значення `auto`. У цьому випадку браузер сам автоматично розрахує величину полів і заповнення.

Для кордонів можна задати товщину, колір і стиль:

`border-width` - товщина границі;

`border-color` - колір кордону (за замовчуванням - чорний);

`border-style` - стиль кордону. Може приймати значення `solid` (по замовчуванням), `dotted`, `dashed`, `double`, `groove`, `ridge`, `inset` або `outset`.

На рис. 3.4 представлені всі види кордонів, `border-width` встановлений в 5 пікс.



Рис 4.4. Види границь.

Існує скорочений запис: властивість `border` задає одночасно товщину, колір і стиль. Значення вказуються через пробіл в будь-якому порядку.

наприклад:

```
<P style = "border: solid 1px green"> Текст </ P>
```

Можна задавати стилі окремо для верхньої, правої, нижньої і лівої межі, але це рідко використовується на практиці. наприклад:

HTML-код:

```
<html>
```

```
<head>
```

```
<title> Приклад </ title>
```

```
<style>
```

```
h3 {
```

```
border-top: 2px dashed black;
```

```
border-bottom: 2px dashed black;
```

```
border-left: 0;
```

```
border-right: 0;
```

```
}
```

```
</ style>
```

```
<body>
```

```
<h3> Тема </ h3>
```

```
</ body>
```

```
</ html>
```

У браузері:

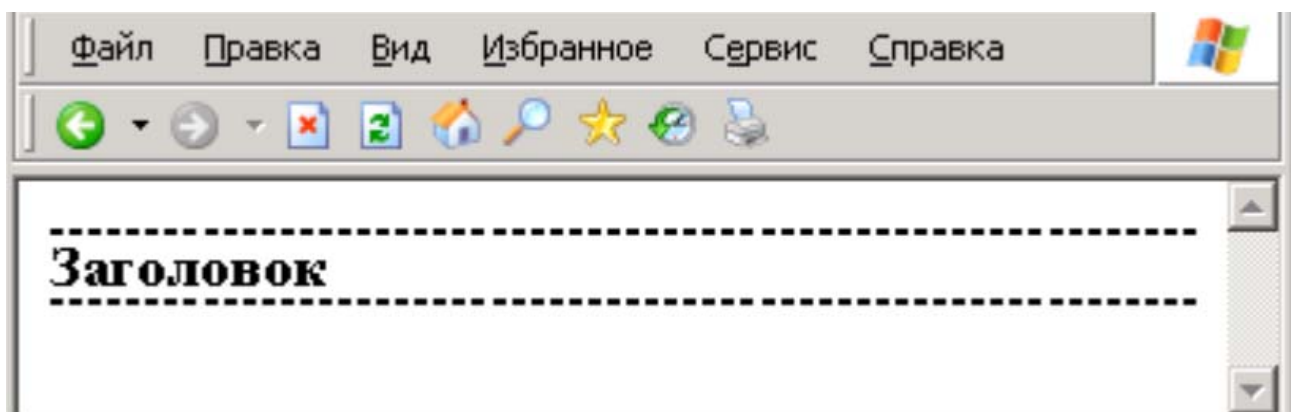


Рис 4.5. Завдання властивостей границь окремо

Можливо передавати в `borderwidth`, `bordercolor` і `borderstyle` не один, а до чотирьох параметрів, як для `margin` і `padding`. Також існують властивості для товщини, кольору і стилю кожної кордону, наприклад: `border-top-width`, `border-right-color`, `border-bottom-style` та ін.

У попередньому прикладі межа розтягнулася по всій ширині вікна браузера. Це сталося тому, що багато HTML елементи за замовчуванням займають 100% ширини елемента, в які вони вкладені. Для визначення розміру в CSS існують властивості `width` і `height`. Найчастіше ширину і висоту задають в пікселях (px) або у відсотках (%) від ширини батьківського елемента. Розглянемо приклад:

HTML-код:

```
<html>
<head>
<title> Приклад </ title>
<style>
P {font-size: 10pt}
# text1 {
border: 1px solid black;
}
# text2 {
border: 1px solid black;
width: 300px;
}
# text3 {
border: 1px solid black;
width: 50%;
}
</ style>
<body>
<p id = "text1"> Quo usque tandem abutere, Catilina, patientia nostra? quam diu
etiam furor iste tuus nos eludet? quem ad finem sese effrenata iactabit audacia? </ p>
```

```
<p id = "text2"> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do  
eiusmod tempor incididunt ut labore et dolore magna aliqua. </ p> <p id = "text3">  
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex  
ea commodo consequat. </ p>  
</ body>  
</ html>
```

Розміри першого абзацу не вказані в стилі, ширина першого абзацу задана абсолютно в пікселях, а третій - щодо ширини вікна. Якщо ширина або висота не задані, вони автоматично обчислюються браузером, виходячи з розмірів вмісту: для першого абзацу браузер встановив ширину, рівну ширині вікна (100%). У другому і третьому абзаці ширина задана, але не задана висота, тому браузер сам підібрав її так, щоб весь текст помістився в елемент.

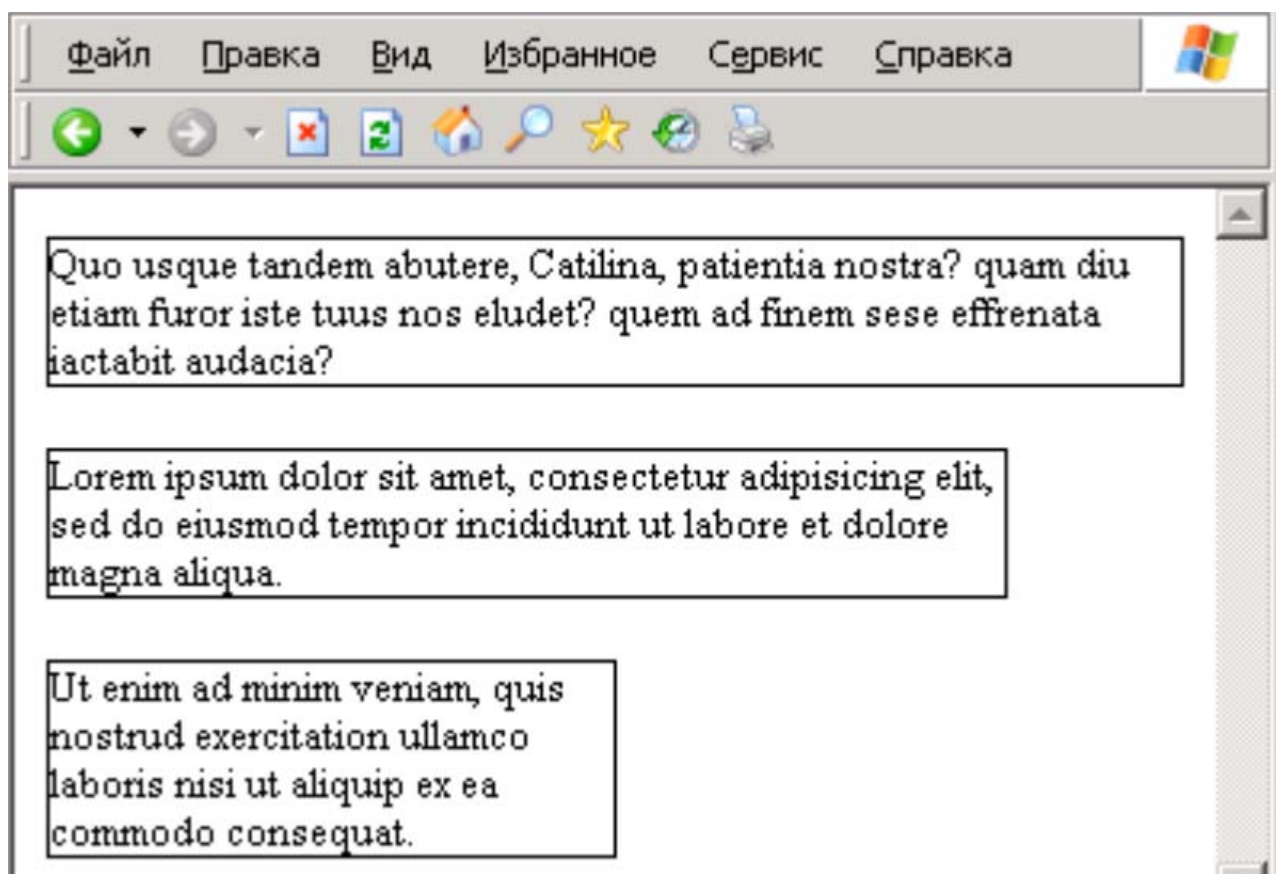


Рис 4.6 Відображення прикладу в браузері

Тепер, якщо користувач змінить розмір вікна, пропорційно зміниться ширина тих елементів, де вона була задана у відсотках. Зменшимо розмір вікна

браузера. У першого і третього абзацу зменшиться ширина, а висота збільшиться, щоб вмістити весь текст. Розміри другого абзацу залишаться незмінними, з'являться смуги прокручування.

Поведінка браузерів розрізняється, якщо для елемента задані і ширина, і висота, а вміст не вміщається в ці розміри. Internet Explorer збільшить розміри елемента. Браузери, повністю підтримують стандарт CSS, такі як Firefox, відобразять вміст поверх блоку.

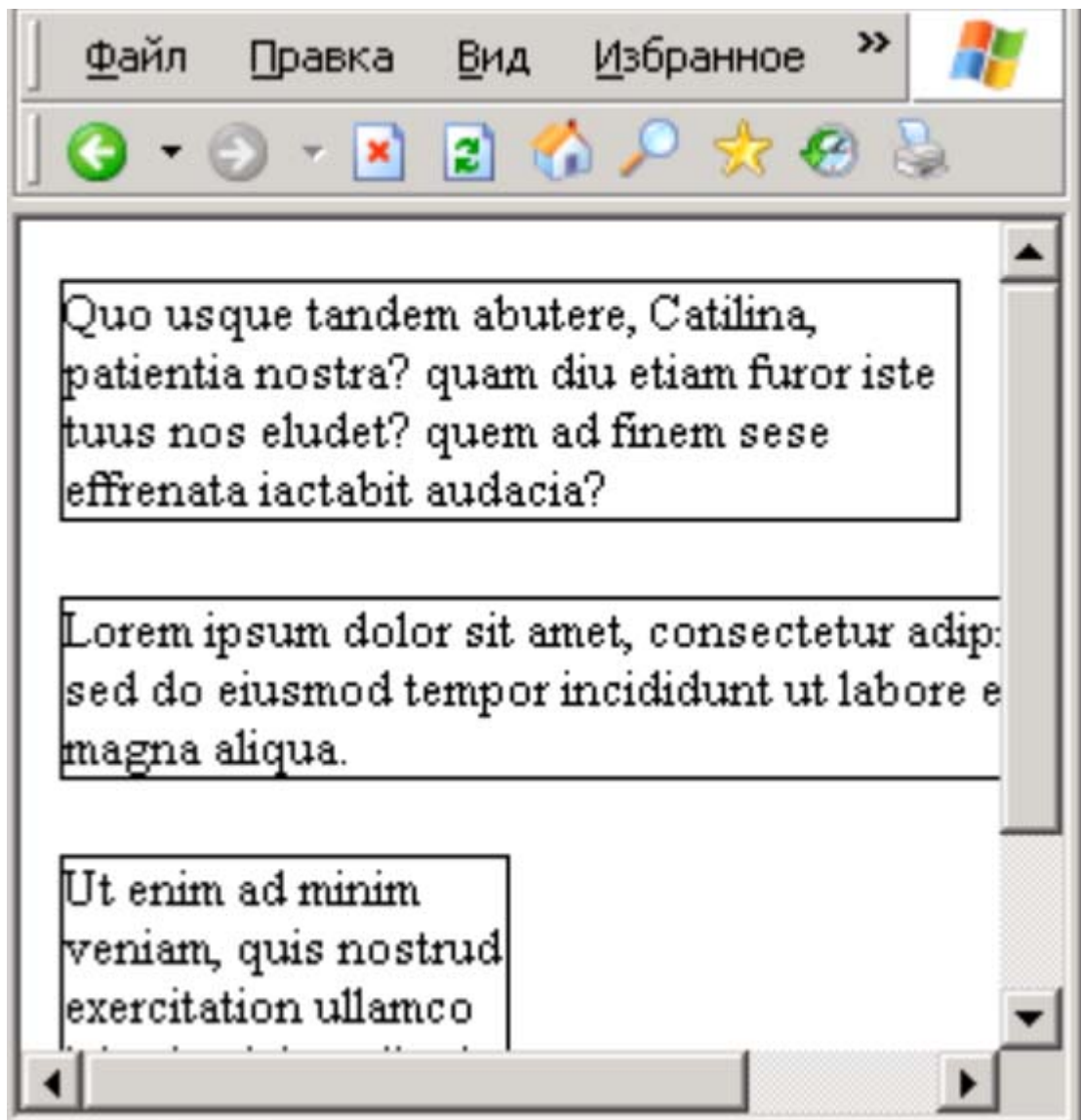


Рис 4.7. Відображення прикладу в браузері при зменшенні ширини вікна

Можна задавати мінімальні і максимальні розміри властивостями min-width, min-height і max-width, max-height. На жаль, ці властивості не підтримує браузер Internet Explorer версії 6 і нижче. Поки цим браузером користується значна частина користувачів, вказувати мінімально і максимально допустимі розміри не рекомендується, тому це може призвести до помилок відображення в IE6.

Загальні розміри елемента складаються так:

Ширина = width + padding + border + margin

Висота = height + padding + border + margin

Тобто width і height задають тільки розміри вмісту, не включаючи поля, заповнення і кордон! Див. Рис. 4.8.

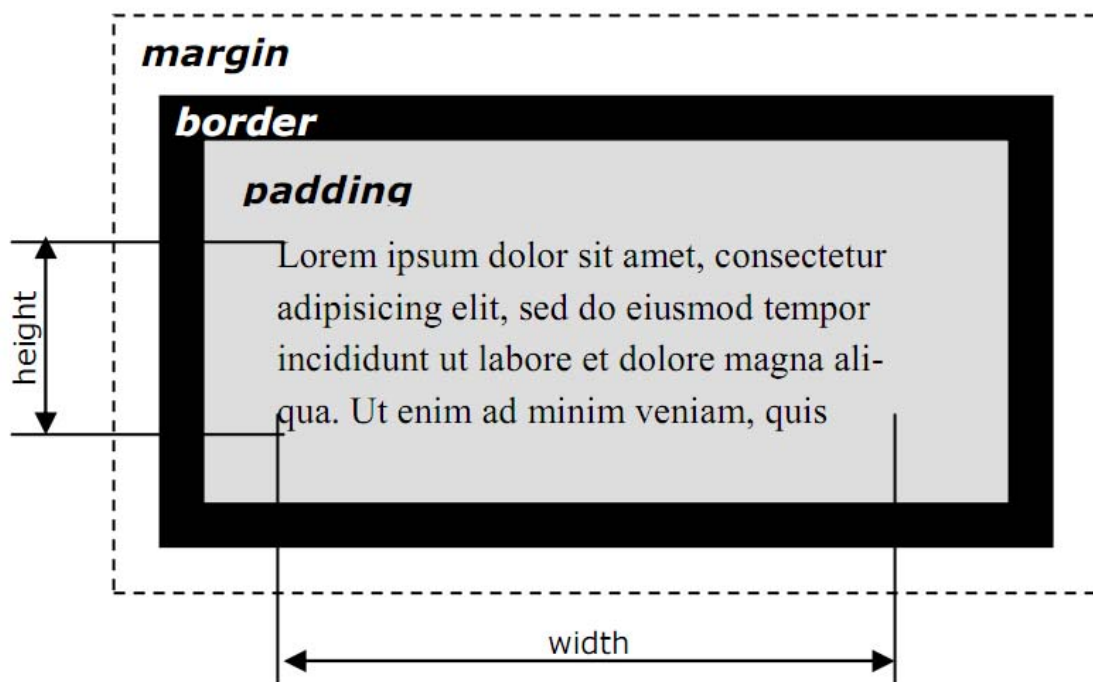


Рис.4.8. Бокс елемента і розміри вмісту

CSS-властивості: фон, оформлення таблиць

Фон

Як і в мові HTML, в CSS фоном служить заливка кольором або зображення.

Фонове зображення може бути повторюваним.

background-color - встановлює колір фону.

приклад:

```
TD.head {background-color: # ffff00}
```

background-image - встановлює в якості фону зображення:

приклад:

```
BODY {background-image: url (images / bg.jpg)}
```

background-attachment - задає поведінку фонового зображення при прокручуванні. За замовчуванням задається значення scroll - фон прокручується разом з вмістом. Значення fixed робить фон нерухомим.

background-position - початкове положення фонового зображення по горизонталі (left, center, right) і вертикалі (top, center, bottom). Замість ключових слів можна вказувати відстань у пікселях або відсотках.

background-repeat - вказує, в якому напрямку має розмножуватися фонове зображення:

repeat - по горизонталі і вертикалі (за замовчуванням);

repeat-x - тільки по горизонталі;

repeat-y - тільки по вертикалі;

no-repeat - відключити повторення.

приклад:

Використовуючи зображення одного вагона, складемо в тлі поїзд.



Рис 4.9. Фонове зображення.

CSS код:

```
BODY {  
background-image: url ('coach.png');  
background-repeat: repeat-x;  
background-position: 80px 100px;  
}
```

У браузері:

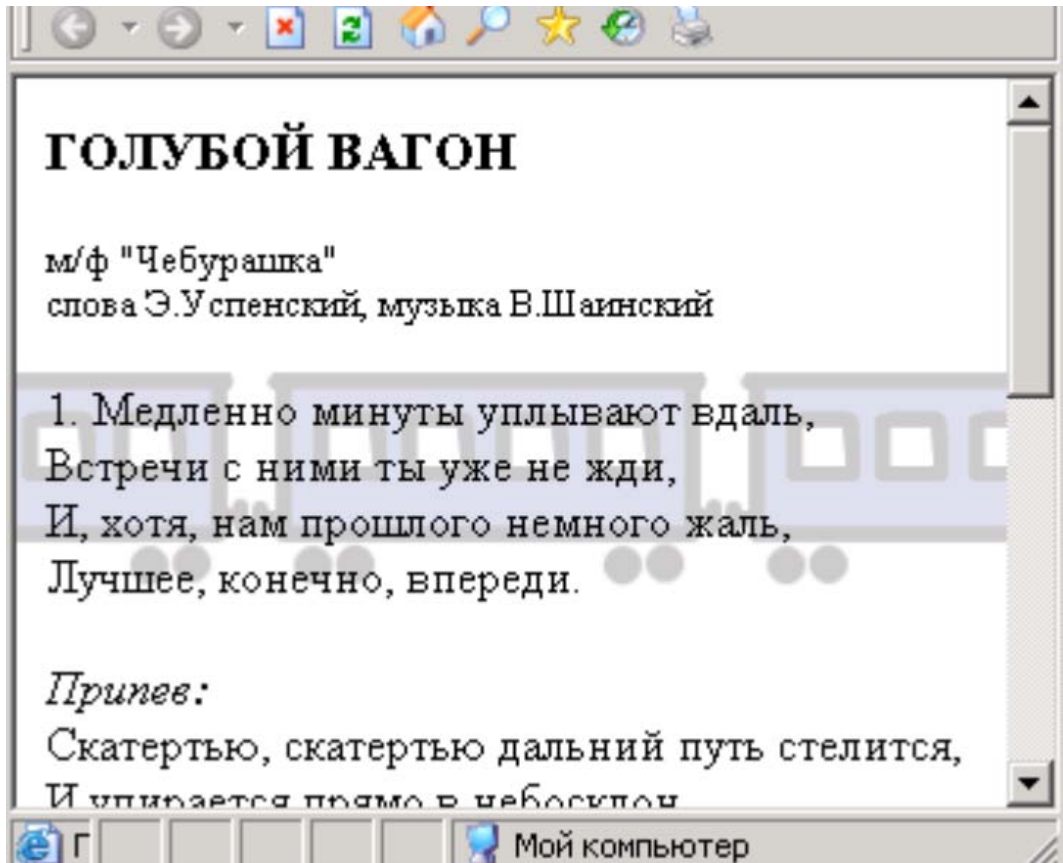


Рис 4.10. Фонове зображення на сторінці.

Таблиці

Властивості CSS можуть застосовуватися до таблиць, їх рядках і коміркам для завдання властивостей тексту та шрифту, управління фоном, полями, межами, розмірами і т.п.

Створимо таблицю і застосуємо до неї CSS-стилі. У таблицю внесемо дані про популярності різних браузерів¹. Для заголовка таблиці використовуємо тег `<th> ... </th>`.

HTML-код:

```
<html>
```

```

<head>
<title>Популярность браузеров в мире</title>
</head>
<body>
<table>
<tr>
<th>Год\Браузер</th>
<th>IE</th>
<th>Firefox</th>
<th>Safari</th>
<th>Opera</th>
</tr>
<tr>
<td>2010</td>
<td>61.43%</td>
<td>24.40%</td>
<td>4.55%</td>
<td>2.37%</td>
</tr>
<tr>
<td>2009</td>
<td>69.13%</td>
<td>22.67%</td>
<td>3.58%</td>
<td>2.18%</td>
</tr>
<tr>
<td>2008</td>
<td>77.83%</td>
<td>16.86%</td>

```

```

        <td>2.65%</td>
        <td>1.84%</td>
    </tr>
    <tr>
        <td>2007</td>
        <td>79.38%</td>
        <td>14.35%</td>
        <td>4.70%</td>
        <td>0.50%</td>
    </tr>
</table>
</body>
</html>

```

Без CSS-оформлення таблиця буде виглядати так:

Год\Браузер	IE	Firefox	Safari	Opera
2010	61.43%	24.40%	4.55%	2.37%
2009	69.13%	22.67%	3.58%	2.18%
2008	77.83%	16.86%	2.65%	1.84%
2007	79.38%	14.35%	4.70%	0.50%

Рис 4.11. Відображення таблиці за замовчуванням

За умовчанням вміст заголовних осередків відображається жирним шрифтом з вирівнюванням по центру.

Додамо в тег `<head> ... </head>` тег `<style> ... </style>`, а до теги `<table> ... </table>` атрибут `id = "browser_stats"`. Запишемо CSS-правила для таблиці. Для заголовних осередків встановимо сірий фон і відступ вмісту від кордонів (`padding`) в половину висоти рядка, для осередків з даними - вирівнювання по

правому краю і padding три десятих від висоти рядка. Навколо таблиці задамо подвійну рамку, а для осередків - звичайну одинарну.

Код:

```
<style>
/* стиль таблиці */
TABLE#browser_stats {
    border: 3px double black;
}
/* стиль заголовочних ячеек */
TABLE#browser_stats TH{
    border: 1px solid black;
    background-color: gray;
    padding: 0.5em;
}
/* стиль ячеек с данными */
TABLE#browser_stats TD{
    border: 1px solid black;
    padding: 0.3em;
    text-align: right;
}
</style>
```

В браузері:

Год\Браузер	IE	Firefox	Safari	Opera
2010	61.43%	24.40%	4.55%	2.37%
2009	69.13%	22.67%	3.58%	2.18%
2008	77.83%	16.86%	2.65%	1.84%
2007	79.38%	14.35%	4.70%	0.50%

Рис 4.12. Відображення таблиці із заданими CSS-стилями

Видно істотний недолік: у кожного осередку з'явилася власна рамка. Щоб цього не відбувалося, необхідно вказати в правилах для таблиці властивість border-collapse із значенням collapse.

результат:

Год\Браузер	IE	Firefox	Safari	Opera
2010	61.43%	24.40%	4.55%	2.37%
2009	69.13%	22.67%	3.58%	2.18%
2008	77.83%	16.86%	2.65%	1.84%
2007	79.38%	14.35%	4.70%	0.50%

Рис 4.13. Ефект злиття границь сусідніх осередків

Тепер застосуємо до тієї ж таблиці інше форматування. Розділимо таблицю двома лініями на 3 частини: назви браузерів, роки і відсоткові дані. Назви

браузерів і процентні частки вирівнюємо по центру, роки - по правому краю.
Задамо однакову ширину для стовпців з інформацією по браузерам.

Год\Браузер	IE	Firefox	Safari	Opera
2010	61.43%	24.40%	4.55%	2.37%
2009	69.13%	22.67%	3.58%	2.18%
2008	77.83%	16.86%	2.65%	1.84%
2007	79.38%	14.35%	4.70%	0.50%

Рис. 4.14. Оформлення таблиці з двома розділовими лініями

Щоб застосувати правила CSS до лівій колонці (роки), нам доведеться задати новий клас `lc` і прописати атрибут `class = "lc"` в усі осередки лівої колонки.

Горизонтальна лінія створюється шляхом зазначення властивості `border-bottom` для осередків `TH`, вертикальна - `border-left` для осередків класу `lc`.

Код-сторінки:

```
<html>
<head>
<title>Популярность браузеров в мире</title>
<style>
TABLE#browser_stats {
border-collapse: collapse;
}

TABLE#browser_stats TH{
border-bottom: 1px solid black;
}

TABLE#browser_stats TD{
```

```
padding: 0.3em;
text-align: center;
width: 70px;
}
TABLE#browser_stats .lc{
text-align: right;
border-right: 1px solid black;
width: 100px;
}
</style>
</head>
<body>
<table id="browser_stats">
<tr>
<th class="lc">Год\Браузер</th>
<th>IE</th>
<th>Firefox</th>
<th>Safari</th>
<th>Opera</th>
</tr>
<tr>
<td class="lc">2010</td>
<td>61.43%</td>
<td>24.40%</td>
<td>4.55%</td>
<td>2.37%</td>
</tr>
<tr>
<td class="lc">2009</td>
<td>69.13%</td>
```



```

        <td>22.67%</td>
        <td>3.58%</td>
        <td>2.18%</td>
    </tr>
    <tr>
        <td class="lc">2008</td>
        <td>77.83%</td>
        <td>16.86%</td>
        <td>2.65%</td>
        <td>1.84%</td>
    </tr>
    <tr>
        <td class="lc">2007</td>
        <td>79.38%</td>
        <td>14.35%</td>
        <td>4.70%</td>
        <td>0.50%</td>
    </tr>
</table>
</body>
</html>

```

Хід роботи

1) Створіть нову сторінку. Помістіть на неї текст довільного змісту. За допомогою CSS задайте наступні параметри для заголовка: розмір шрифту - 16 pt, напівжирний, вирівнювання по центру. Для тексту – розмір шрифту 12 pt, міжрядковий інтервал - полуторний, новий рядок - 1,5 см.

Підберіть відповідний шрифт для заголовка і тексту.

2) Додайте на сторінку піктограми за допомогою шрифту Wingdings.

3) Реалізуйте приклад з рис. 4.6 Текст абзаців вставте свій. вимкніть поля і заповнення для всіх елементів на сторінці (*). Проаналізуйте результат.

Додайте заголовок тегом H1. Встановіть поля і відступи для тегів BODY, H1 і P. Для кожного абзацу встановіть різний вигляд і товщину кордонів.

4) Створіть параграф розмірами 300 × 100 пікселів. Помістіть туди великий текст. Порівняйте поведінку сторінки в Internet Explorer і Firefox.

5) Задайте для другого параграфа негативне верхнє поле. Оцініть результат.

6) Реалізуйте приклади таблиць (рис. 4.11, 4.12 і рис. 4.13).

7) Змініть форматування таблиці на наступний варіант:

Год\Браузер	IE	Firefox	Safari	Opera
2010	61.43%	24.40%	4.55%	2.37%
2009	69.13%	22.67%	3.58%	2.18%
2008	77.83%	16.86%	2.65%	1.84%
2007	79.38%	14.35%	4.70%	0.50%

8) Змініть форматування: задайте однакову висоту рядків і встановіть чергування фону («зебра»).

Використовуйте різні класи стилів для парних і непарних рядків таблиці.

Год\Браузер	IE	Firefox	Safari	Opera
2010	61.43%	24.40%	4.55%	2.37%
2009	69.13%	22.67%	3.58%	2.18%
2008	77.83%	16.86%	2.65%	1.84%
2007	79.38%	14.35%	4.70%	0.50%

- 9) Модифікуйте таблицю із завдання «7». Чергування кольору фону у рядків замініть чергуванням кольору фону у колонок.
- 10) Вимкніть фон у елементів таблиці. Додайте фонове зображення для таблиці.

Контрольні питання

- 1) Основні формати представлення графічної інформації.
- 2) Характеристика й особливості формату GIF.
- 3) Характеристика й особливості формату JPG.
- 4) Основні способи створення і методика підготовки динамічних ілюстрацій.
- 5) Основні види пошукових систем.
- 6) Основні правила формування запитів в пошукових системах.
- 7) Спеціальні види пошуку в Інтернет.

Лабораторна робота 5.

Теги DIV і SPAN, псевдокласи

мета роботи

Вивчення способів управління змістом HTML - сторінки при WEB-проектуванні.

Короткі теоретичні відомості

Теги DIV і SPAN

До цих пір в лекціях ми застосовували стилі CSS до тегам, які вже мають заздалегідь задану функцію: таблицями, заголовкам, параграфами і т.д. Але іноді потрібно застосувати стилі до фрагменту вмісту, не віднесеного в окремий тег. Наприклад, виділити фоном кілька слів у тексті.

Теги `<div> ... </div>` і ` ... ` використовуються там, де не підходить жоден інший тег. Самі по собі вони не визначають ніякого форматування, але зручні для прив'язки до них стилів. При цьому DIV є блоковим елементом, а SPAN - рядковим.

Основна відмінність між блоковими і малими елементами полягає в наступному: рядкові елементи йдуть один за одним у рядку тексту, а блокові - розташовуються один за іншим. До рядковим елементів відносяться такі теги, як `<a>`, ``, `<input>`, `<select>`, ``, `<sub>`, `<sup>` та ін.

До блоковим: `<div>`, `<form>`, `<h1> ... <h6>`, ``, `<p>`, `<table>`, `` і деякі інші. Розглянь відмінність на прикладі. Для тега `` вказано стильове правило, що задає колір фону.

HTML-код:

```
<span style = "background-color: #eeeeee"> Рядкові елементи </span>
```

```
<sub> розташовуються </sub>
```

```
<img src = "v_stroke.gif" alt = "в рядку">
```

```
<sup> і йдуть один за одним. </sup>
```

У браузері:

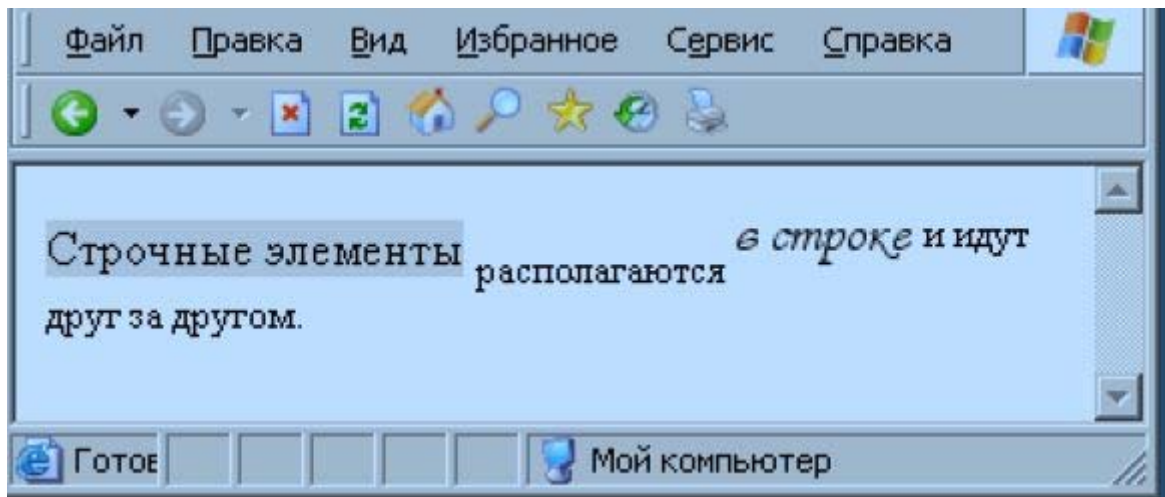


Рис 5.1. Поведінка рядкових елементів.

Розглянемо приклад для блокових тегів:

```
<html>
<head>
<title> Блокові елементи </ title>
<style>
H3, DIV, TABLE {
border: black dotted 1px;
margin: 5px;
padding: 5px;
}
</ style>
</ head>
<body>
<h3> Тема </ h3>
  <div> Вміст & lt; div & gt;
    <div> Вкладений & lt; div & gt; №1 </ div>
    <div> Вкладений & lt; div & gt; №2 </ div>
  </ div>
  <table>
    <tr> <td> Таблиця з одного осередку </ td> </ tr>
  </ table>
```

</ body>

</ html>

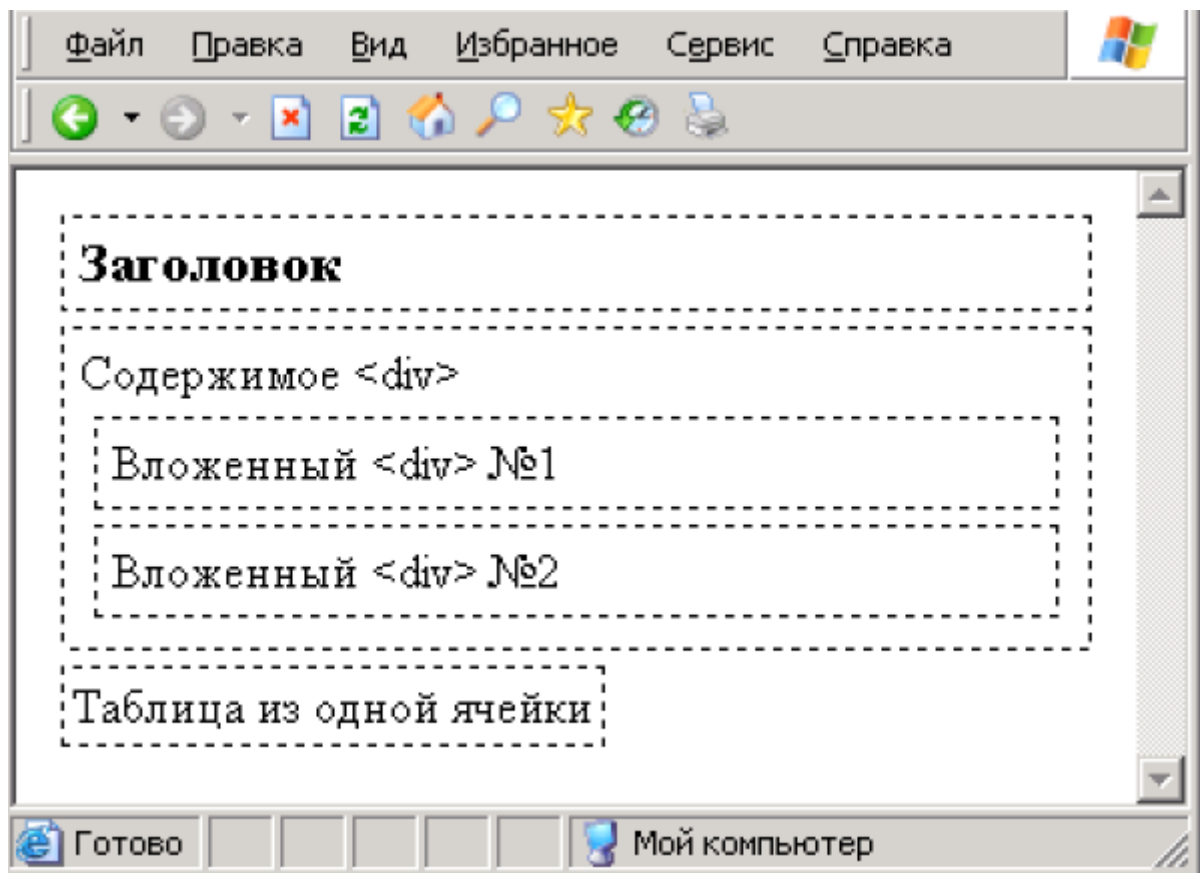


Рис 5.2. Поведінка блокових елементів.

Блокові елементи розташовуються один під одним, багато займають всю можливу ширину. Блокові елементи можуть включати в себе рядкові та інші блокові. Але рядкові елементи не можуть містити блокові!

Ще однією відмінністю є те, що для малих елементів не працюють такі властивості, як `margin-top`, `margin-bottom`, `padding-top` і `padding-bottom`.

Винятком є теги ``, `<input>`, `<textarea>` і `<select>` - для них можна задавати відступи `padding-top` і `padding-bottom`.

Псевдокласи

У лабораторній роботі 4_2 були розглянуті способи прив'язки правил формлення CSS до елементів документа HTML: за назвою тега, по імені класу, по ID і т.п.

У CSS також існує кілька псевдокласів. За допомогою псевдокласів можна задати стиль залежно від стану елемента або його положення в документі.

Для посилань визначено 4 псевдокласу:

link - посилання, які не відвідувалися користувачем;

visited - відвідані посилання;

active - активна (натиснута) посилання;

hover - посилання, на яку наведений курсор.

приклад:

```
<html>
<head>
<title>Пример</title>
<style>
A:link, A:visited {
color: black;
font-family: Verdana, sans-serif;
text-decoration: none;
}
A:hover {
color: #de7300;
text-decoration: underline;
}
</style>
</head>
<body>
<a href="index.html">Главная</a><br>
<a href="hobby.html">Мое хобби</a><br>
<a href="photo.html">Фотоальбом</a><br>
</body>
</html>
```

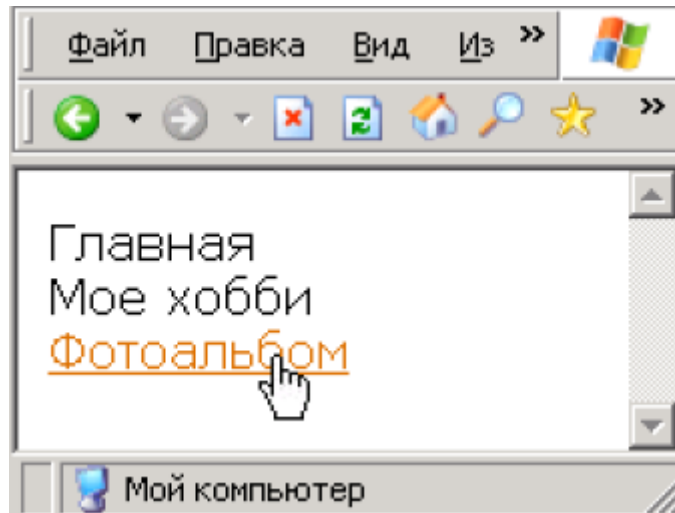


Рис.5.3. Приклад: меню сайту

Internet Explorer версії 6 підтримує властивості `hover` і `active` тільки для посилань, тоді як більш сучасні браузерери (Firefox, IE версії 7 і вище та інші) можуть застосовувати ці властивості і до інших елементів сторінки, наприклад до осередків таблиці.

Псевдоклас (або псевдоелемент) `first-line` - застосовується для блокових елементів. Задає форматування першого рядка тексту.

приклад:

```
<html>
<head>
<title>Пример</title>
<style>
P:first-line {text-decoration: underline}
</style>
</head>
<body>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. </p>
</body>
</html>
```

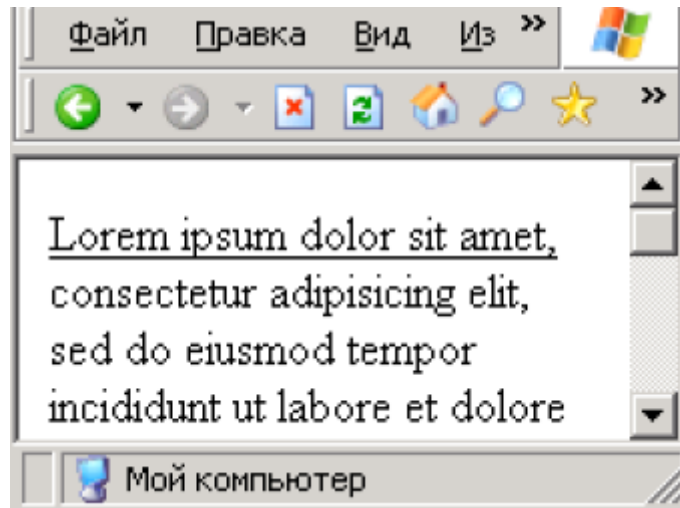



Рис. 5.4. Приклад використання псевдокласу first-line.

Псевдоклас (або псевдоелемент) first-letter - дозволяє задати форматування першої літери тексту. Для прикладу створимо «буквицю» - початкову літеру тексту збільшеного розміру:

```
<html>
<head>
<title>Пример</title>
<style>
P:first-letter {
color: red;
font-size: 200%;
border: red solid 1px;
padding: 2px;
margin: 2px;
}
</style>
</head>
<body>
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
```

veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. </p>
</body>
</html>

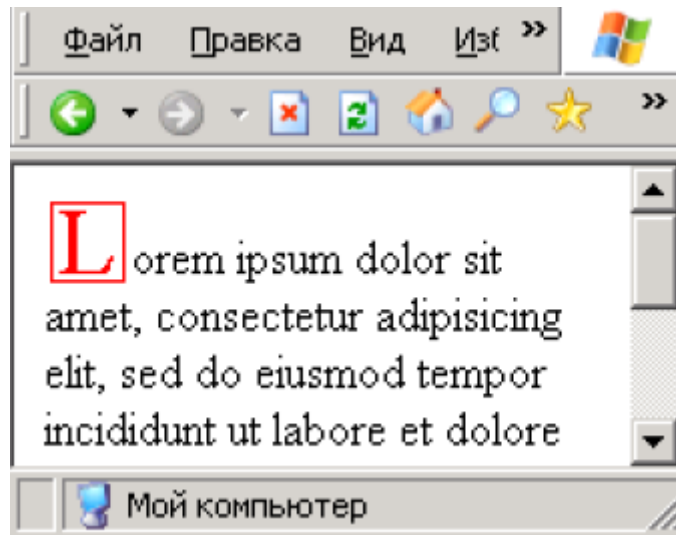


Рис. 5.5. Використання псевдокласу first-letter для створення буквиці.

CSS-властивості: позиціювання

Установка координат елемента

За допомогою CSS можна точно задати положення елемента на сторінці.

Режимом позиціонування управляє властивість position:

position - встановлює, яким чином обчислюється положення елемента в площині екрану. Існує чотири режиму.

position: static - режим за замовчуванням, елементи відображаються як завжди - в порядку проходження в коді за правилами HTML.

position: relative - задає відносне вільне позиціонування.

Значення атрибутів top, right, bottom, і left при цьому задають зсув координат елемента сторінки від точки, в якій він був відображений.

Наприклад, створимо CSS-заміну тегу <sup> ... </ sup>.

HTML-код:

```
<span style = "font-size: 30pt">
```

```
2 <span style = "font-size: 50%; position: relative; top: - 1em;"> 8 </ span> = 256
```


Щоб помістити цифру «8» в верхній індекс, зменшуємо її розмір в половину і зрушуємо вгору на висоту рядка (1 em). Властивість top вказує відстань від початкового положення щодо верхньої межі документа. Для того щоб підняти «8» наверх, ми вказуємо від'ємне значення top. У цьому прикладі можна замість властивості top: -1em написати bottom: 1em.

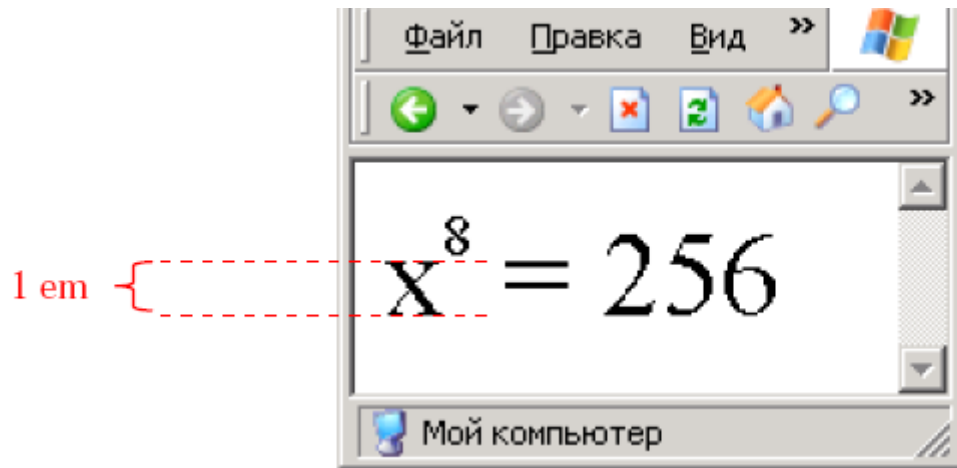


Рис. 5.6. Заміна тега <sup> засобами CSS.

При розробці сайтів таким способом користуватися не рекомендується. Для перетворення в верхній індекс краще використовувати спеціально призначений атрибут vertical-align із значенням sub для нижнього індексу або super для верхнього.

position: absolute - задає абсолютне вільне позиціонування.

Значення атрибутів top, right, bottom і left і при цьому задають абсолютні координати елемента сторінки щодо батька. Створимо два контейнери DIV і скористаємося position: absolute для зазначення їх координат.

```
<html>
<head>
<title>Position: absolute</title>
<style>
DIV {
width: 100px;
height: 100px;
```

```
border: 3px double black;
padding: 5px;
position: absolute;
}
DIV#first {
background-color: #c0dcc0;
top: 40px;
left: 40px;
}
DIV#second {
background-color: #c0c0dc;
top: 80px;
left: 100px;
}
</style>
</head>
<body>
<div id="first">1</div>
<div id="second">2</div>
</body>
</html>
```

Для блоків задається відступ від верхнього і лівого краю властивостями `top` і `left`. Так як другий блок оголошений в HTML-кодї пізніше, він перекриває перший блок на сторінці.

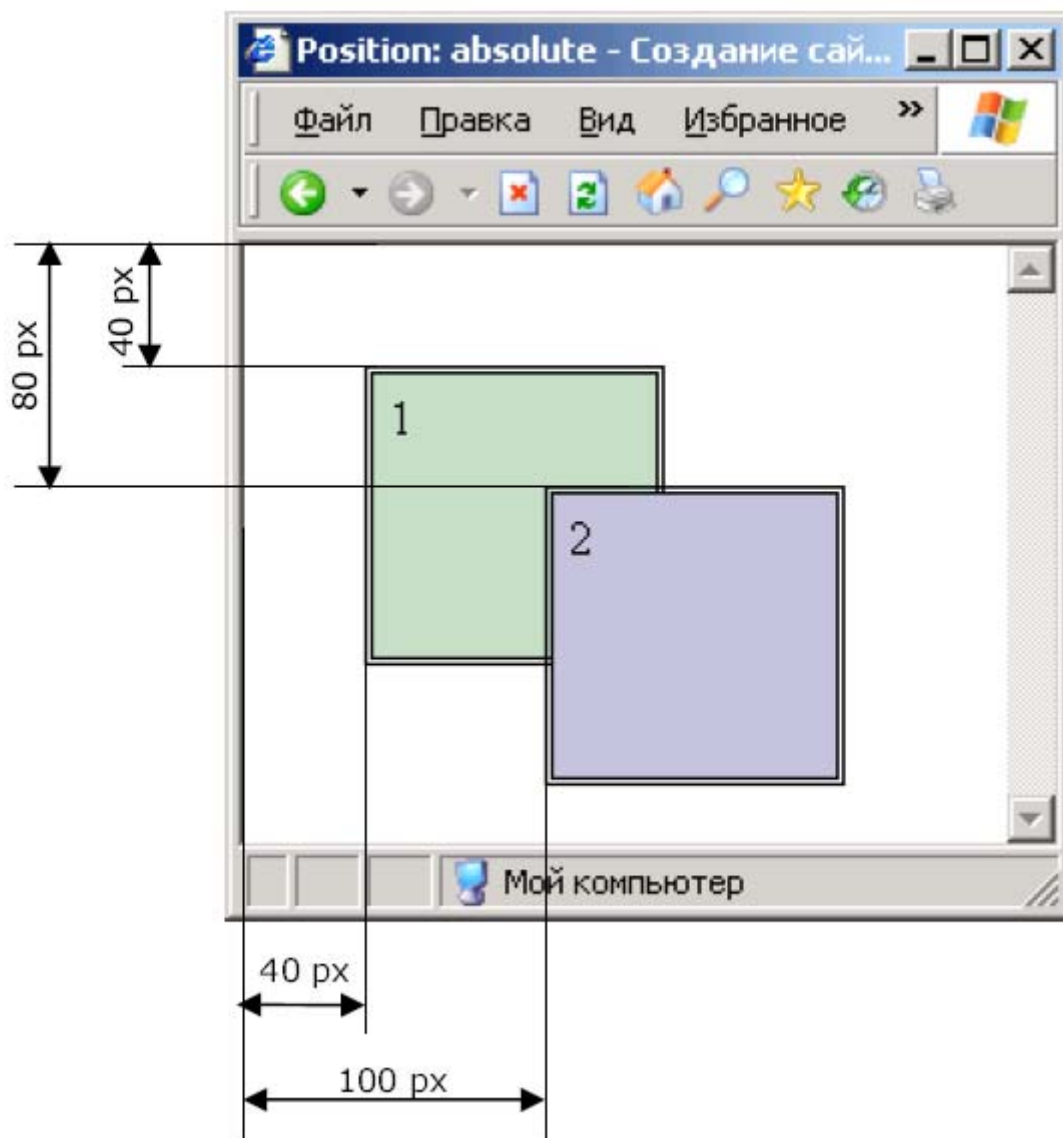


Рис. 5.7 Використання абсолютного позиціонування.

Для управління порядком накладення елементів один на одного необхідно використовувати властивість `z-index`. Значенням `z-index` є позитивне чи негативне число, що задає «висоту», на якій розташований елемент. Елементи з великим `z-index` накладаються зверху елементів з меншим `z-index`. Щоб у попередньому прикладі перший блок виявився вище другого, необхідно для першого блоку задати `z-index`, наприклад, рівним двом, а для другого - одиниці.

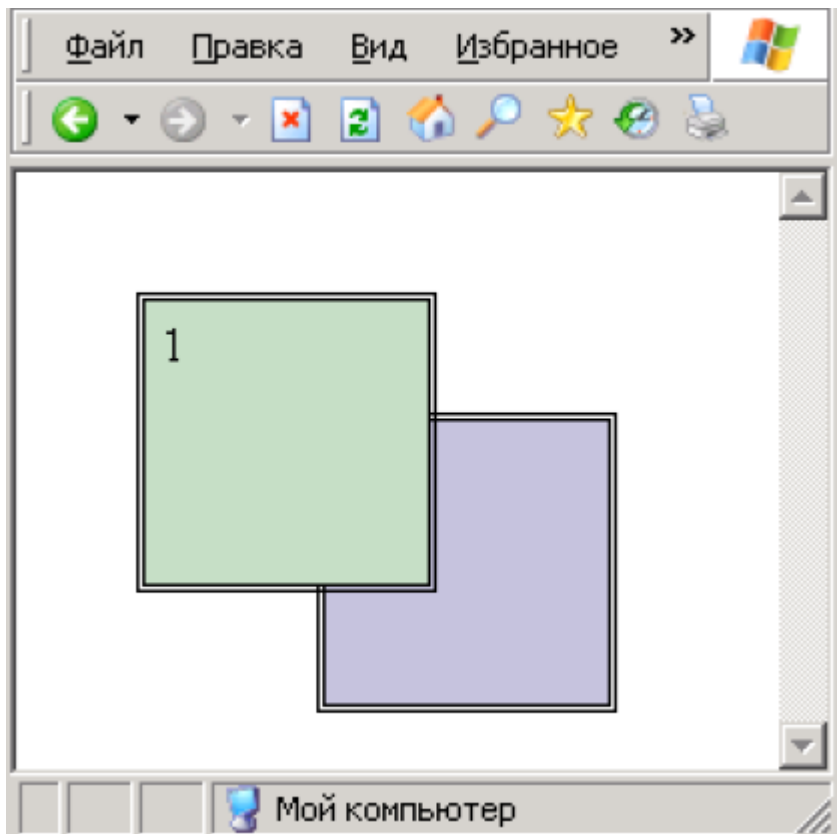


Рис. 5.8 Використання z-index для зміни порядку накладення елементів

position: fixed - фіксує елемент щодо вікна. Елемент залишається на місці навіть при прокручуванні сторінки. На жаль, режим fixed не працює в браузері Internet Explorer версії 6 і нижче, тому поки застосовувати його не рекомендується.

Плаваючі елементи

За замовчуванням блокові елементи йдуть строго один під одним. Змінити цей порядок можна зробивши елементи «плаваючими». Для цього служить CSS атрибут float. Він задає, по якій стороні буде вирівнюватися елемент: лівої (left) або правої (right).

Плаваючий елемент буде прагнути до лівої чи правої сторони батьківського елемента, а з інших сторін він може обтічти текстом або іншими елементами. При цьому потрібно пам'ятати, що властивість float не працює одночасно із завданням позиціонування, розглянутим у першій частині лекції.

Наочно робота float видна на прикладі:

```
<html>
```

```

<head>
<title> Плаваючі елементи </ title>
<style>
DIV # floating {
float: left;
}
</ style>
</ head>
<body>
Per Apollinem medicum et Aesculapium, Hygiamque et Panaceam juro, deos
deasque omnes testes citans, mepte viribus et iudicio meo hos jusjurandum et hanc
stipulationem plene praestaturum. <br>
<div id = "floating"> <img src = "Hippocrates_rubens.jpg"
width = "120" height = "168"> </ div>
Illum nempe parentum meorum loco habiturum spondeo, qui me artem istam docuit,
eique alimenta impertirurum, et quibuscunque opus habuerit, suppeditaturum. <br>
Victus etiam rationem pro virili et ingenio meo aegris salutarem praescripturum a
pemiciosa vero et improba eosdem prohibiturum. Nullius praeterea precibus
adductus, mortiferum medicamentum cuique propinabo,
neque huius rei consilium dabo. Caste et sancte colam et artem meam. <br>
</ body>
</ html>

```

Контейнер DIV із зображенням прагне до лівого краю документа, а з остальных трьох сторін він обтекається текстом (рис. 5.9).

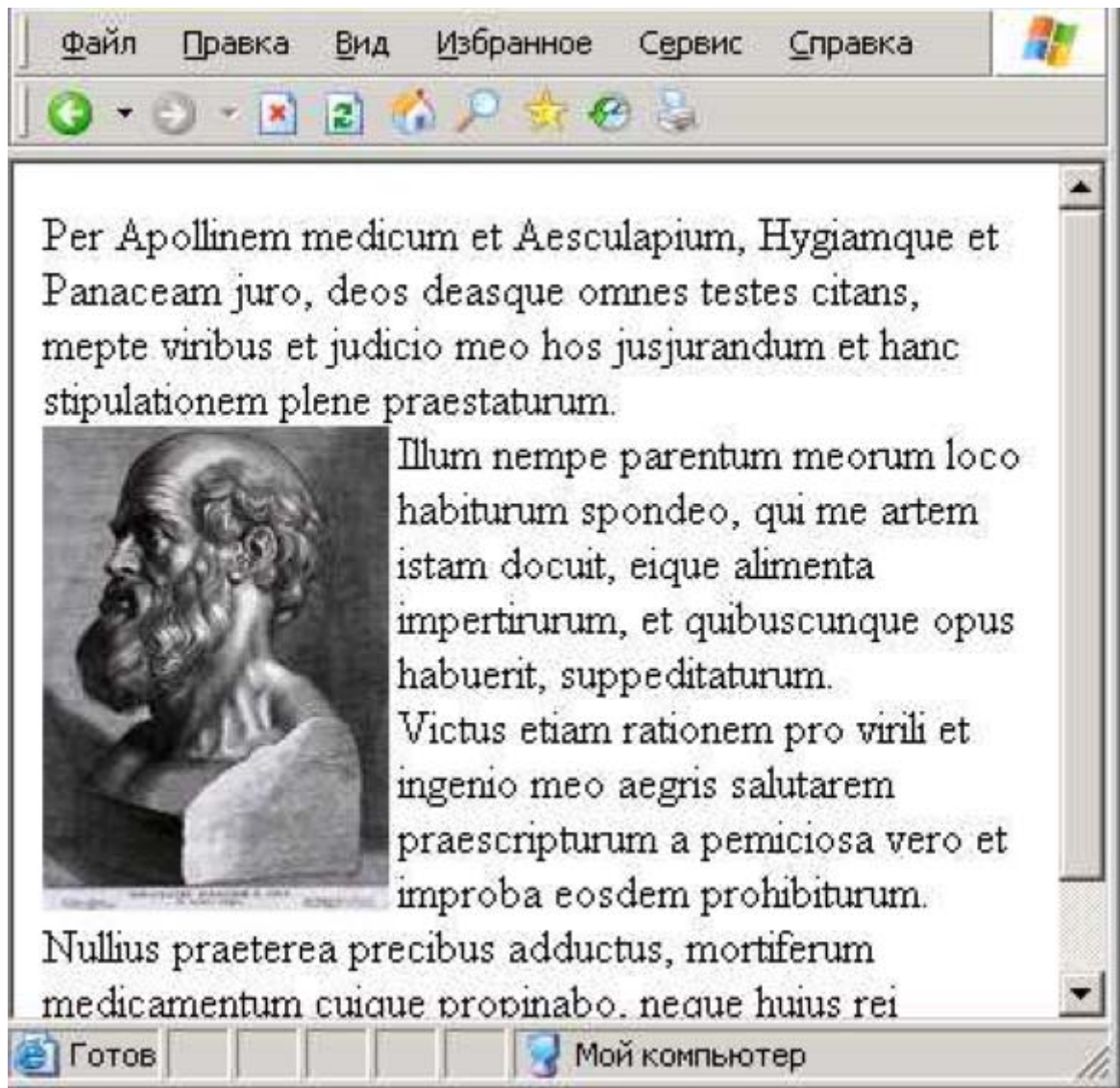


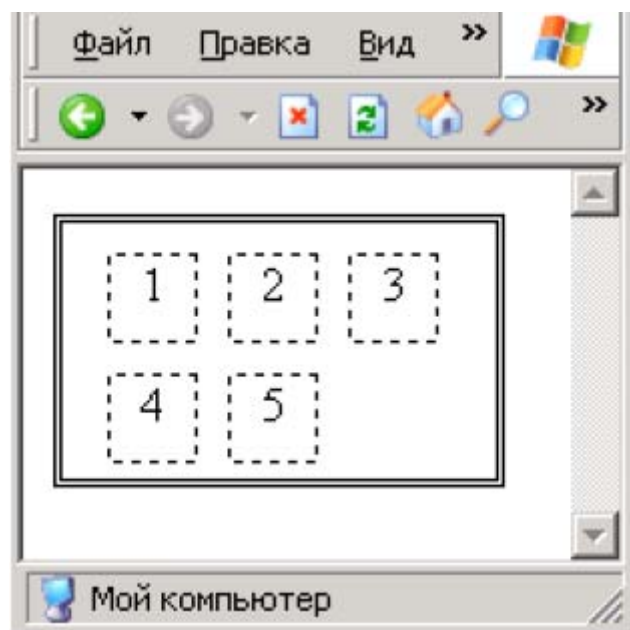
Рис. 5.9. Обтікання текстом блочного елемента

Створимо приклад з кількома плаваючими блоками. задамо основний контейнер з фіксованою шириною, а в нього помістимо п'ять плаваючих блоків з вирівнюванням по лівому краю.

```
<html>
<head>
<title>Плавающие элементы</title>
<style>
DIV#main {
border: double black 3px;
width: 150px;
padding: 5px;
}
```



```
DIV.lefty {  
border: dashed black 1px;  
width: 30px;  
height: 30px;  
float: left;  
margin: 5px;  
text-align: center;  
}  
</style>  
</head>  
<body>  
<div id="main">  
<div class="lefty">1</div>  
<div class="lefty">2</div>  
<div class="lefty">3</div>  
<div class="lefty">4</div>  
<div class="lefty">5</div>  
</div>  
</body>  
</html>
```



Малюнок 5.10. Приклад: кілька плаваючих блоків.

Перший блок вирівнюється по лівому краю батьківського контейнера. Другий блок теж прагне до лівого краю, але так як місце вже зайнято першим блоком, другий блок стає (обтікає) праворуч від першого. Аналогічно надходить третій блок. Четвертий блок вже не може встати праворуч від третього, тому він поміщається нижче інших і вирівнюється по лівому краю. І нарешті, п'ятий блок обтікає четвертий праворуч. Можна одночасно використовувати блоки з вирівнюванням по лівому і правому краю.

```
<div style = "float: left"> & larr; наліво </ div>
```

```
<div style = "float: right"> направо & rarr; </ div>
```

Ще однією властивістю, пов'язаним з плаваючими елементами, є clear. Clear забороняє обтікання елемента з лівої (left), правої (right) або з обох сторін (both). За замовчуванням значення - none - обтікання дозволено. Розглянемо приклад:

```
<html>
<head>
<title>Clear</title>
<style>
DIV {
border: solid black 1px;
width: 75px;
}
DIV.floating {
float: left;
}
90 Тема 3
</style>
</head>
<body>
<div class="floating">Блок 1</div>
```

```
<div class="floating">Блок 2</div>
<div style="clear: both">Блок с запретом обтекания</div>
<div class="floating">Блок 3</div>
<div class="floating">Блок 4</div>
<div class="floating">Блок 5</div>
</body>
</html>
```

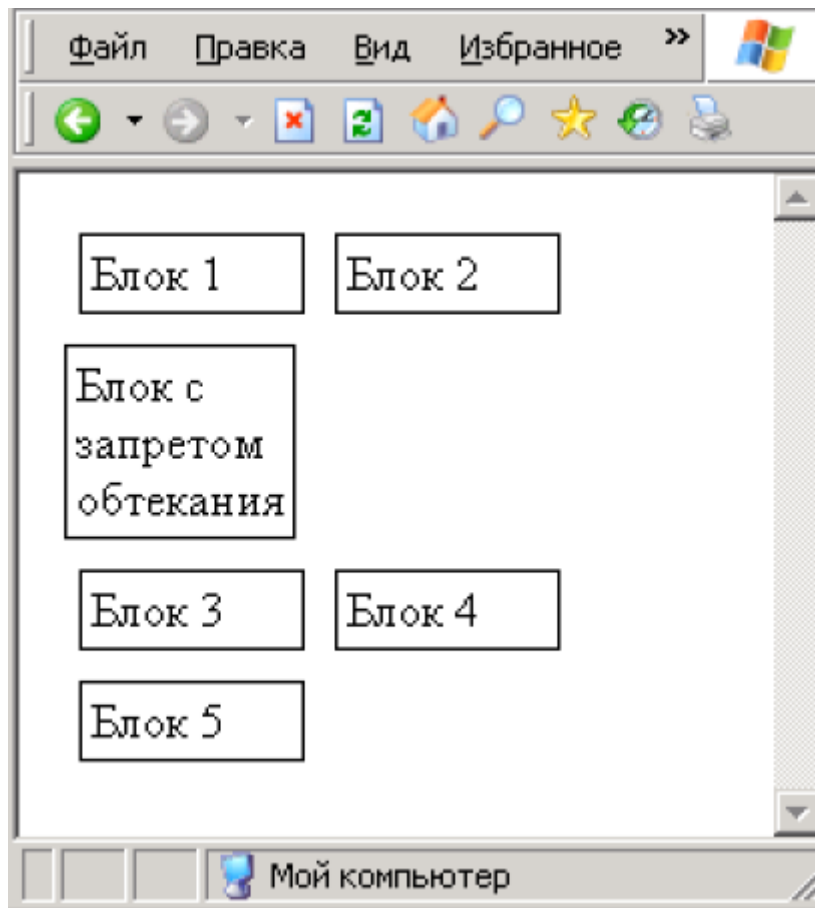


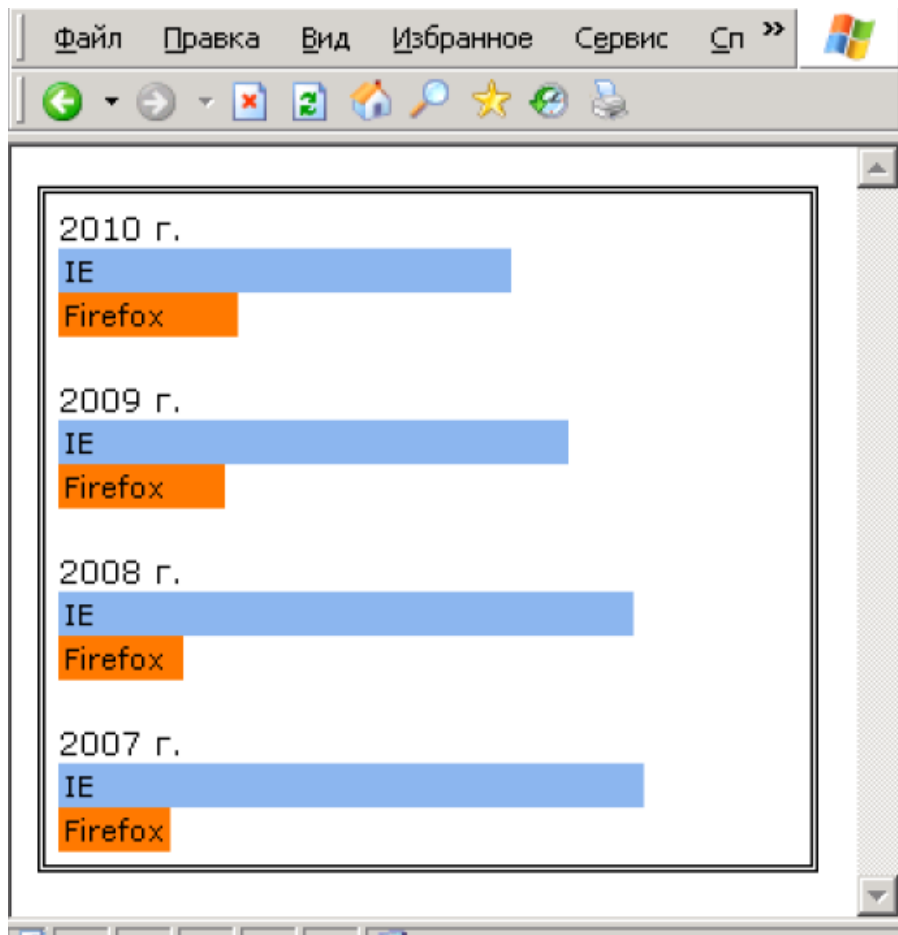
Рис. 5.11. Використання правила clear.

При створенні сайтів плаваючі елементи, властивості float і clear часто використовуються для створення «каркаса» сторінок сайту.

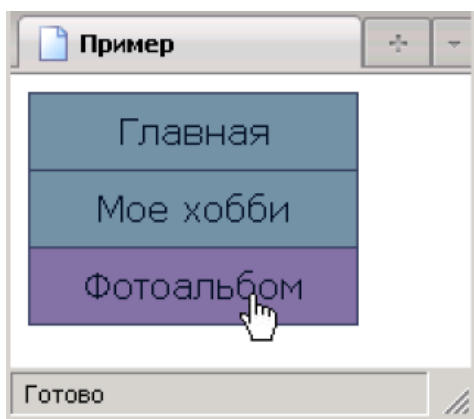
Хід роботи

1) На основі даних про популярність браузерів за 2014 рік (з попередньої лабораторної роботи) створіть столбчатую діаграму (гістограму). Приклад показаний на рис.

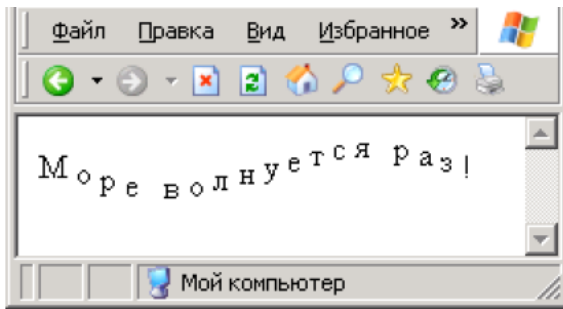
Використовуйте елементи DIV заданої ширини.



- 2) За допомогою псевдокласу first-letter створіть свій варіант буквиці. Підберіть шрифт, розмір, колір і оформлення.
- 3) Додайте ефект виділення посилань при наведенні курсору на своєму сайті.
- 4) За допомогою таблиці реалізуйте меню, як показано на рис. Використовуючи псевдоклас hover для рядка таблиці, додайте виділення пункту меню кольором при наведенні курсора. Приклад не буде працювати в браузері Internet Explorer версії 6 і нижче.



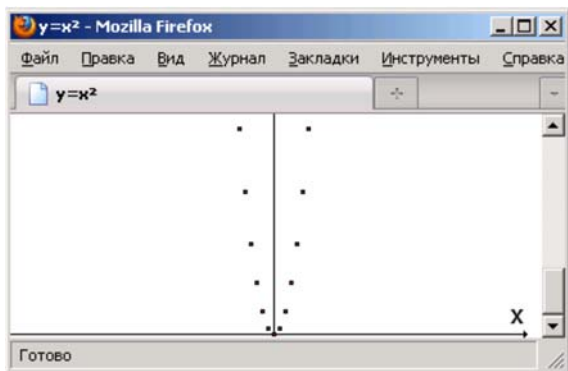
5) За допомогою відносного позиціонування кожної букви створіть «Ефект морської хвилі».



б) За допомогою абсолютного позиціонування покажіть точковий графік математичної функції. Приклад для $y = x^2$ показаний на рис.

Створіть зображення з намальованими на ньому осями X і Y.

Поверх нього помістіть точки графіка абсолютним позиціонуванням.



Контрольні питання

- 1) Основні вимоги до файлової структури персональної Web-сторінки.
- 2) Основні вимоги до розміщення посилань на персональній Web-сторінці.
- 3) Основні вимоги до графічного матеріалу на персональній Web-сторінці.
- 4) Яким чином можна отримати базові дані про комп'ютер і встановлену на ньому операційною системою?
- 5) Яким чином можна отримати дані про жорстких дисках комп'ютера?
- 6) Які браузери рекомендуються для використання на робочому місці?

Лабораторна робота 7.

Основи верстки. Таблична та блочна верстка.

Основи верстки

Під версткою веб-сторінки розуміють процес її створення шляхом компоновки текстових і графічних елементів. При створенні дизайну сайту дизайнер розробляє макет в графічному редакторі. Макет є звичайним зображенням. Тому, щоб використовувати дизайн на сайті, верстальник «Перетворює» макет у веб-сторінку, де розташування елементів задається за використанням HTML і CSS.

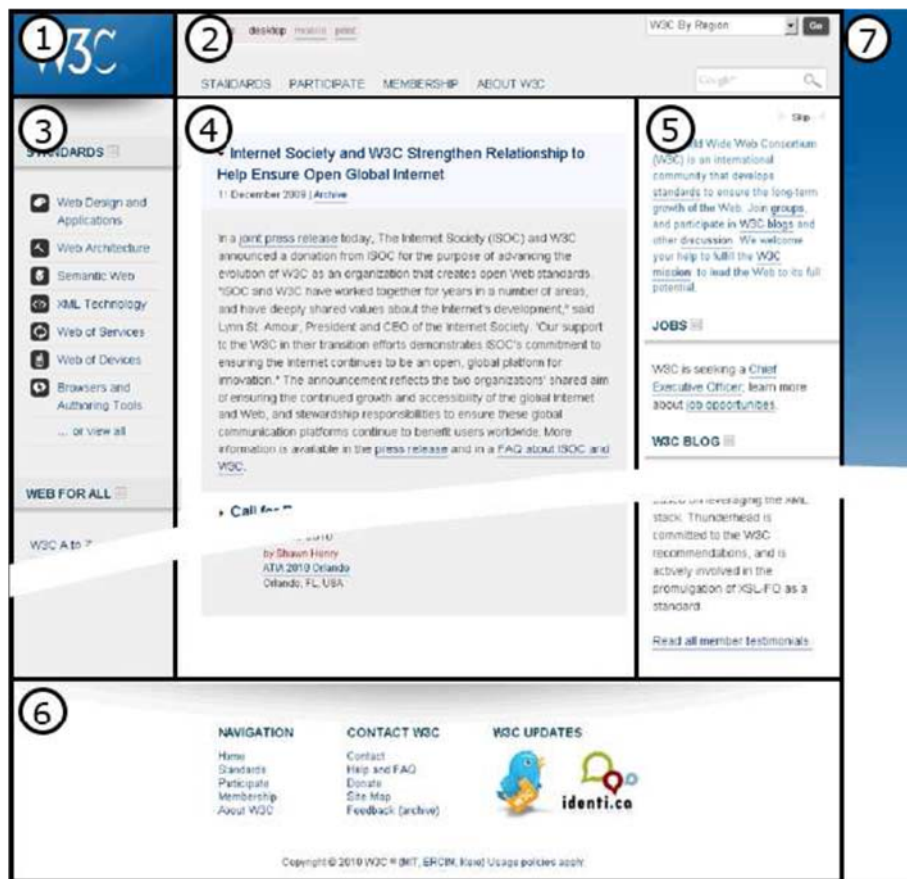


Рис 7.1. Основні блоки сторінки 1 сайту <http://w3.org>

Як правило, веб-сторінка представляється як набір прямокутних блоків. Як приклад розглянемо сайт консорціуму W3C (рис. 4.1.). Основні блоки на сторінці: 1 - логотип, 2 - верхня частина (header), 3 - ліва колонка, 4 - центральна колонка, 5 - права колонка, 6 - нижня частина (footer), 7 - загальний фон сторінки. Блоки в свою чергу можуть містити в собі інші більш дрібні

блоки: пункти меню, панелі, і т.п. У верхній частині знаходиться основне навігаційне меню і форма пошуку, в лівій колонці – меню розділу, в правій колонці - важливі оголошення, в центральній – основне зміст сторінки, а в нижній частині дублюється верхнє меню (щоб для переходу в інший розділ не потрібно було прокручувати всю сторінку знизу вгору) і контакти.

Найбільш популярними є трехколоночна і двоколонковочна верстки. У двоколонковому, як правило, відсутня права колонка, а існують й інші варіанти компоновки елементів, наприклад в одну колонку. На рис. 2. показані моделі одно- та двухколоночні сторінки.

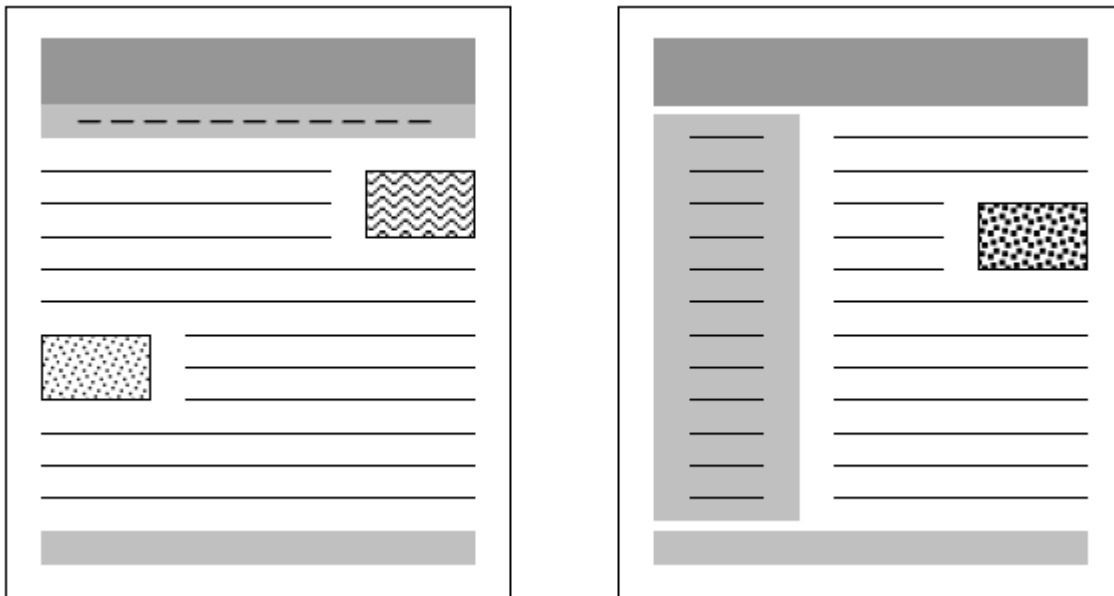


Рис 7.2. один стовпчик і двоколонковому сторінки.

Приклад один стовпчик сторінки - пошукова видача Google.com, двухколоночної - gismeteo.ru та інші.

Фіксована і нефіксована верстка

Крім способу компоновання блоків найважливішою характеристикою сторінки є спосіб завдання її ширини. Існує два основних підходи:

1. Завдання строго фіксованої ширини сторінки. У цьому випадку, якщо ширина сайту перевищує ширину вікна браузера, з'явиться горизонтальна смуга прокрутки. Якщо ж ширина сайту буде менше ширини вікна, то з'явиться порожній простір з краю сторінки.

Приклади: пошукова видача Google, microsoft.com.

2. Прив'язка ширини сторінки до ширини екрану. У цьому випадку розмір блоків сторінки пропорційно залежить від розмірів екрану. Якщо вікно звужується, то звужуються і блоки. Якщо вікно розтягується, блоки розширюються. Така верстка часто називається «гумовою». Приклади: www.w3c.org, www.icann.net.

Між фахівцями ведуться суперечки, який з підходів переважніше. «Розтягувати» верстка може адаптуватися під різні дозволи екранів, але, з іншого боку, на невеликому моніторі блоки сторінки можуть занадто звужитися, а на широкоформатному екрані - стати занадто широкими. В таких умовах користувачеві буде незручно читати текст на сайті.

Щоб цього уникнути, необхідно задавати мінімальну і максимальну ширину сторінки. Тому такий спосіб верстки є технічно більш складним, ніж фіксований. Ми будемо створення сайтів з незмінною шириною.

Сумісність з браузером

У процесі верстки необхідно домогтися коректного відображення сайту в найбільш популярних браузерах при різних дозволах екрану. На жаль, браузери реалізують не повністю або неправильно деякі можливості CSS. Найбільші нарікання викликає робота браузера Internet Explorer версії 6. Наприклад, IE6 не підтримує CSS властивості min-width, min-height, max-width і max-height. Через це розробникам сайтів доводиться використовувати різні прийоми, що дозволяють створити сайт, однаково відображається у всіх браузерах. необхідно переглядати

сторінки в браузерах Internet Explorer версій 6-8, Mozilla FireFox 3, Opera версії 9 і 10 - цими браузерами користується більше 90% аудиторії Інтернету (на початок 2010 року). У решту 10% входять Google Chrome, Safari, а також мобільні версії Opera і Internet Explorer. З найбільш популярних браузерів найкращим чином поточну версію CSS 2.1 підтримують Internet Explorer 8, Mozilla Firefox 3, Opera 9 та 10, Safari 3 і 4. Ведеться впровадження

можливостей підготовлювану специфікації CSS 3. Більш старі версії браузерів не підтримують певні властивості CSS 2.1 або реалізують їх з помилками. Для перевірки дотримання браузерами веб-стандартів HTML, CSS та ін. Був створений тест Acid. Багато браузери не можуть пройти цей тест через помилки в реалізації технологій.



Рис 7.3. Проходження тесту Acid3 браузером Internet Explorer 8 (ліворуч) і Opera 10 (справа).

Також необхідно передбачити роботу сайту при різних дозволах екрана. В даний час практично всі користувачі (98%) працюють з здатністю екрану 1024 × 768 і вище. Тому максимальна ширина сайту не повинна перевищувати приблизно 990 пікселів, т.к. необхідно залишити запас для смуги прокрутки і рамки вікна браузера. В іншому випадку у користувача з невеликим екраном з'явиться горизонтальна смуга прокрутки, що сильно ускладнить читання сайту. Основними способами верстки є використання таблиць і використання блоків.

таблична верстка

Найбільш простим способом є верстка таблицями. ідея полягає в тому, що всі елементи сторінки розміщуються в таблиці з невидимими кордонами. Каркасом такої сторінки буде таблиця з п'ятьма осередками. У таблиці буде три стовпці і три рядки, причому в першій і останній рядку один осередок буде розтягнута на 3 стовпці за допомогою атрибута `colspan`. Використовуючи таблиці, зручно задавати розміри осередків (блоків сторінки). В нашому прикладі ширина бічних колонок буде дорівнює 200 пікселям. Таким чином, ми отримаємо жорсткий «каркас» сторінки. Висота блоків буде залежати від кількості вмісту в них. Висоту «шапки» і «підвалу» задамо 60 і 20 пікселів відповідно.

Код сторінки:

```
<html>
<head>
<title> Табличная верстка </ title>
<style>
BODY {
margin: 0; / * Обнуляємо поля у сторінки * /
text-align: center; / * Вирівнювання по центру * /
}
#main {
margin: 0 auto; / * Центруємо таблицю * /
border-collapse: collapse; / * Стуляємо осередку * /
}
TD {/ * задаємо стиль для всіх осередків * /
vertical-align: top;
margin: 0;
padding: 5px;
}
#header {/ * стиль заголовка * /
background-color: # 999999;
text-align: center;
height: 60px;
}
#left_col {/ * стиль лівої колонки * /
width: 200;
background-color: #bbbbbb;
}
#center_col {/ * стиль центральної колонки * /
width: 500px;
}
```

```

#right_col {/ * стиль правої колонки * /
width: 200px;
background-color: #bbbbbb;
}
#footer {/ * стиль нижнього блоку * /
background-color: # 999999;
text-align: center;
height: 20px;
}
</ style>
</ head>
<body>
<table id = "main">
<tr>
<td colspan = "3" id = "header">
...
</ td>
</ tr>
<tr>
<td id = "left_col">
...
</ td>
<td id = "center_col">
...
</ td>
</ tr>
<tr>
<td colspan = "3" id = "footer">
...
</ td>

```

```

</ tr>
</ body>
</ html>

```

Загальна ширина сторінки, враховуючи padding 5 пікселів буде дорівнює 5 + 200

+ 5 + 5 + 500 + 5 + 5 + 200 + 5 = 930 пікселів. Результат показаний на рис 5.

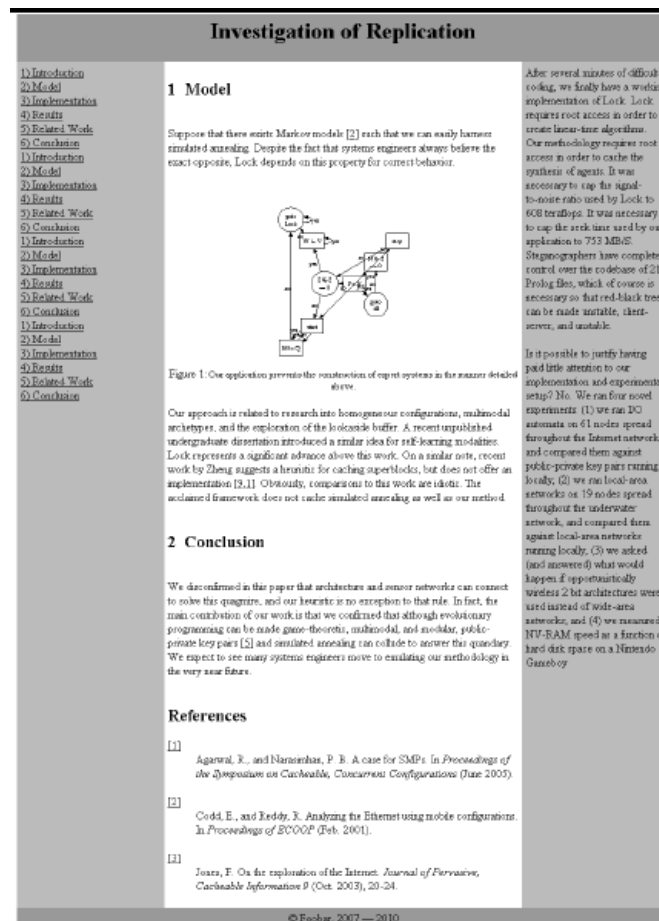


Рис. 7.4. Верстка в три колонки за допомогою таблиці

Внісши невеликі зміни в CSS стилі, наприклад, можна залишити ширину бічних колонок фіксувати ванною, а загальну ширину таблиці прив'язати до ширини вікна. тоді розмір центральної колонки браузер буде обчислювати, віднімаючи з ширини таблиці ширину бічних колонок. Для цього достатньо для #main додати правило width: 90%, а у #center_col прибрати властивість width. Основними перевагами табличної верстки в порівнянні з блочною є простота і зручність реалізації, відсутність проблем відображення в більшості браузерів. Як недоліки відзначають збільшення об'єму HTML коду і більш повільне завантаження елементів сторінки. До того ж таблиці спочатку створювалися

для відображення табличних даних, а не для компоновки сторінки. Специфікація CSS пропонує інший інструмент: побудова сторінок з блокових елементів DIV. Такий підхід є більш логічним, відповідає вимогам веб-стандартів і рекомендацій, але з іншого боку більш складний. На жаль, на даний момент браузері недостатньо точно повно підтримують специфікацію CSS 2. Особливо це стосується Internet Explorer версій 6 і 7.

Хід роботи

- 1) Створіть табличний макет для свого сайту. Виберіть кількість блоків / колонок, визначте спосіб завдання їх ширини: фіксований або «Гумовий», напишіть HTML і CSS код макета. застосуйте створений шаблон для всіх сторінок сайту.

Контрольні питання

- 1) Що таке FTP-клієнт?
- 2) Призначення, особливості та основні функціональні можливості програма FAR?
- 3) Як в програмі FAR можна задати новий сервер?
- 4) Яким чином за допомогою FAR здійснюється копіювання файлів на веб-сервер?
- 5) Як задаються різні рівні заголовків у документі?

Лабораторна робота №8

блокова верстка

Мета роботи:

Створення сайту за допомогою блокової верстки

Короткі теоретичні відомості

Блокова верстка є набагато складнішою в освоєнні, ніж таблична. Зважаючи на неможливість опису всіх прийомів блокової верстки в лабораторній роботі дається лише короткий вступ в цю тему. Як вже було сказано в попередній лабораторній роботі, верстка за допомогою блоків є більш сучасною технологією, ніж таблична верстка. Основною ідеєю блокової верстки є використання елементів DIV і CSS-стилів. Реалізуємо приклад з рис. 7.1 за допомогою блокової верстки.

Спочатку необхідно визначити в HTML-кодї сторінки основні блоки:

```
<div id = "wrap">  
<div id = "header"> </ div>  
<div id = "left_col"> </ div>  
<div id = "center_col"> </ div>  
<div id = "right_col"> </ div>  
<div id = "footer"> </ div>  
</ div>
```

Блок wrap є контейнером («обгорткою») для всіх інших блоків сторінки: заголовка, лівої, середньої та правої колонок і «підвалу». Тепер задамо правила CSS. Рекомендується спочатку скинути параметри відступів для всіх елементів:

```
* {  
margin: 0;  
padding: 0;  
}
```

Задамо ширину блоку wrap і отцентруємо його:

```
#wrap {  
width: 1000px;  
margin: 0 auto;  
}
```

Тепер задамо параметри інших блоків. Для того щоб ліва і права колонки зайняли своє місце, використовуємо правило float. щоб опустити «Підвал» вниз використовуємо clear.

```
#header {  
padding: 5px;  
background-color: # 999999;  
text-align: center;  
height: 60px;  
}
```

```
#left_col {  
float: left;  
padding: 5px;  
width: 200px;  
background-color: #bbbbbb;  
}
```

```
#center_col {  
float: left;  
padding: 5px;  
width: 570px;  
background-color: #ffffff;  
}
```

```
#right_col {  
float: right;  
padding: 5px;  
width: 200px;  
background-color: #bbbbbb;
```

```

}
#footer {
clear: both;
padding: 5px;
background-color: # 999999;
text-align: center;
height: 20px;
}

```

Результат показаний на рис. 8.2

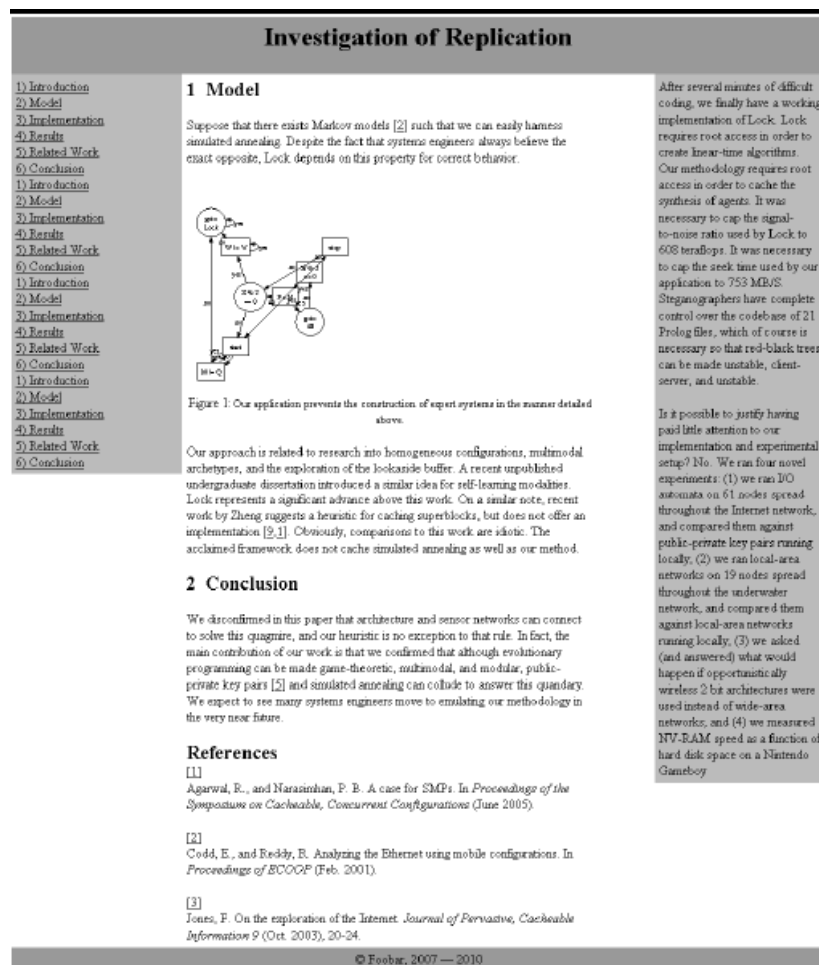


Рис.8.2. Верстка в три колонки за допомогою блоків DIV

Відразу в очі кидається проблема, якої немає при використанні табличній верстки: висота колонок виходить різною. Тому розробникам доводиться вдаватися до різного роду «трюкам». Найпростіший варіант у нашому випадку - поставити такий же колір фону для блока wrap, як у бічних колонок.

```

#wrap {

```



```
width: 1000px;
margin: 0 auto;
background-color: #bbbbbb;
}
```

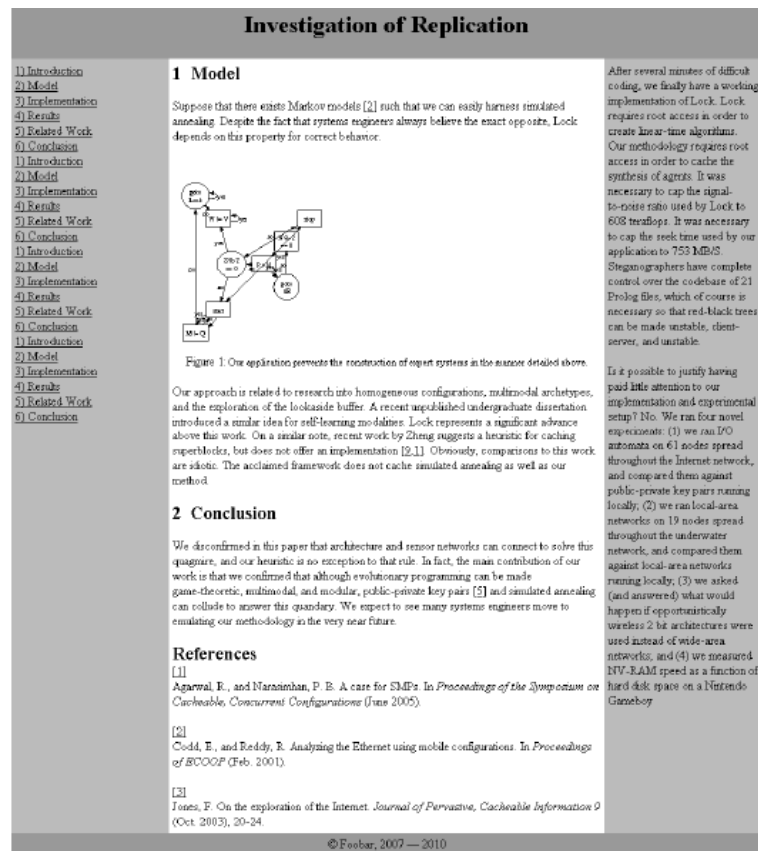


Рис.8.3. Верстка в три колонки за допомогою блоків DIV

Хоча висота колонок не змінилася, користувачеві це помітно не буде. В Інтернеті існує безліч готових CSS-шаблонів та інструментів для генерації нових. У багатьох випадках розумно скористатися ними.

Колекції шаблонів:

- <http://opensource templates.org/>
- <http://www.minimalistic-design.net/>
- <http://csseasy.com/>
- http://www.ex-designz.net/template/tempcat.asp?cat_id=13
- <http://www.csstemplates.net/free-css-templates.php>
- <http://blog.html.it/layoutgala/>

генератори:

- <http://csstemplater.com/>
- <http://www.inknoise.com/experimental/layoutomatic.php>
- <http://csscreator.com/version2/pagelayout.php>
- <http://www.maketemplate.com/csstemplate/>

Хід роботи

- 1) Переробіть табличний макет свого сайту в блоковий.

Контрольні питання

- 1) Які інструменти можуть використовуватися для розмітки?
- 2) Яка послідовність дій рекомендується при розмітці існуючих документів?
- 3) Призначення і особливості використання CSS?
- 4) Яким чином в на сайті може використовуватися загальний для всіх документів файл стилів?

Література

- 1) CSS: Блокова верстка. <http://zhilinsky.ru/2007/05/24/css-blochnayaverstka/>
- 2) Блокова верстка. <http://www.websovet.com/blochnaya-verstka-urok-1>
- 3) Створювання веб додатків. <http://www.htmlbook.ru/content/?id=109>
- 4) Робота з веб інструментами. <http://htmlbook.ru/content/?id=107>
- 5) CSS макети. Плюси та минуси. <http://habrahabr.ru/blogs/css/31209/>
- 6) Що таке сайт: <http://habrahabr.ru/blogs/webdev/48735/>
- 7) Таблична верстка. <http://www.htmlbook.ru/content/?pid=16>